

*Trabajo Fin de Grado*

# Descubrimiento guiado del entorno físico de la biblioteca de la UC3M mediante una aplicación Android cliente-servidor

---

Grado en Ingeniería en Tecnologías de la  
Telecomunicación  
2017-2018

Alumno:  
Víctor Seoane Mérida

Tutora:  
Iria Manuela Estévez Ayrés

Leganés, Junio de 2018  
Universidad Carlos III de Madrid  
Escuela Politécnica Superior



## Resumen

Este trabajo fin de grado consiste en el diseño e implementación de una aplicación para el sistema operativo Android que permite descubrir de forma guiada la biblioteca de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid a través de un juego interactivo.

La aplicación interactúa con el usuario a través de preguntas, enigmas y códigos QR repartidos por el entorno físico de la biblioteca que deben ser escaneados por el usuario, ofreciendo una experiencia personalizada en función de las características del alumno y los espacios que ya ha visitado.

La aplicación ha sido diseñada mediante el entorno de desarrollo Android Studio, que ofrece útiles herramientas para el diseño de aplicaciones móviles de forma sencilla y visual.

Con el fin de ofrecer una experiencia a medida, este trabajo ha requerido de la implementación de un servidor Java mediante Apache, así como de una sencilla base de datos en MySQL, que almacena el progreso de los distintos usuarios en el proceso de descubrimiento de la biblioteca.

El principal objetivo de este trabajo es dotar a los nuevos alumnos, así como a los alumnos que no conocen al completo la biblioteca, de un medio interactivo de conocer todas las posibilidades que este organismo ofrece: préstamos, aulas, servicios, información, etc.

*“Caminando en línea recta no puede  
uno llegar muy lejos.”*

Antoine de Saint-Exupéry.



## Agradecimientos

Nadie dijo que fuera fácil. Han sido cuatro años muy intensos, pero también muy felices. Aún recuerdo el primer día que pisé el auditorio de esta universidad, lleno de miedos y de nervios. ¡Cómo ha cambiado todo desde entonces!

En primer lugar, agradecer a mi madre y a mi padre su infinita paciencia. Gracias por la oportunidad de ir a la universidad, por aguantarme en las épocas de exámenes y por saberos este trabajo de memoria. Sin vosotros, nada hubiera sido posible.

A mis amigos, por haber sido la parte fundamental de esta etapa tan bonita que siempre recordaré. Por el cariño, la confianza, las palabras de ánimo. Porque no hay nada más bonito que sentirse identificado.

A mi tutora Iria, por darme la oportunidad de realizar este trabajo con ella y hacerme sentir especial como alumno desde el principio, nunca ser uno más.

A Rosa Sánchez, directora de la biblioteca de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, por proporcionarme el apoyo y la ayuda necesarios para llevar a cabo una prueba experimental en la biblioteca.

A los profesores Carlos Alario y Jesús Arias, por la ayuda desinteresada proporcionada durante el período de baja de mi tutora Iria.

A mis compañeros de Erasmus, la pieza clave de la mejor experiencia que he vivido estando en esta universidad. Gracias por enseñarme a frenar, a disfrutar y a darme cuenta de que hay tiempo para todo.

A mi familia, por ser el ancla de mi vida, ser el lugar a donde uno siempre puede volver. Gracias por haberme hecho sentir siempre tan arropado.

A todas las personas que, sin saberlo, formaron una pequeña parte de todo esto, que un día se cruzaron en mi camino y me hicieron ser como soy ahora. Gracias, de corazón.

## Tabla de contenido

|                                                          |     |
|----------------------------------------------------------|-----|
| Resumen.....                                             | III |
| Agradecimientos .....                                    | V   |
| Tabla de contenido .....                                 | VI  |
| Índice de Figuras .....                                  | IX  |
| Índice de tablas .....                                   | XI  |
| 1. Introducción.....                                     | 1   |
| 1.1. Motivación.....                                     | 1   |
| 1.2. Objetivos.....                                      | 1   |
| 1.3. Marco regulador .....                               | 2   |
| 1.4. Estructura de la memoria .....                      | 3   |
| 2. Estado del Arte.....                                  | 5   |
| 2.1. Introducción .....                                  | 5   |
| 2.2 Trabajos previos .....                               | 5   |
| 2.2.1. Universidad Pompeu Fabra .....                    | 5   |
| 2.2.2. Universidad de Ryerson y Museo de Arte Inuit..... | 6   |
| 2.2.3. Otros usos documentados.....                      | 7   |
| 2.3. Tecnologías en el lado del cliente .....            | 8   |
| 2.3.1. Plataforma .....                                  | 8   |
| 2.3.1.1. iOS.....                                        | 8   |
| 2.3.1.2. Android .....                                   | 10  |
| 2.3.1.3. Web.....                                        | 10  |
| 2.3.1.4. Elección final.....                             | 11  |
| 2.3.2. Entorno de desarrollo integrado.....              | 11  |
| 2.3.2.1. Android Studio .....                            | 11  |
| 2.3.2.2. NetBeans .....                                  | 13  |
| 2.3.2.3. Elección final.....                             | 13  |
| 2.4. Tecnologías en el lado del servidor .....           | 14  |
| 2.4.1. Lenguajes de programación .....                   | 14  |
| 2.4.1.1. Python.....                                     | 15  |
| 2.4.1.2. Java .....                                      | 15  |
| 2.4.1.3. Elección final.....                             | 15  |
| 2.4.2. Apache Tomcat.....                                | 16  |
| 2.5 Base de datos .....                                  | 16  |
| 2.6. Comunicación Cliente-Servidor .....                 | 18  |
| 2.6.1. Volley .....                                      | 18  |
| 2.6.2. JSON .....                                        | 18  |
| 2.6.3. Inicio de sesión .....                            | 19  |
| 2.6.3.1. Google Sign-In .....                            | 19  |

|                                                                |    |
|----------------------------------------------------------------|----|
| 2.6.3.2. Facebook Login.....                                   | 20 |
| 2.6.3.3. Métodos propios.....                                  | 20 |
| 2.6.3.4. Elección final.....                                   | 21 |
| 2.7. Códigos QR .....                                          | 21 |
| 3. Diseño.....                                                 | 24 |
| 3.1. Requisitos .....                                          | 24 |
| 3.1.1. Requisitos del servidor.....                            | 24 |
| 3.1.2. Requisitos del cliente .....                            | 24 |
| 3.1.3. Requisitos de la base de datos .....                    | 24 |
| 3.2. Diseño del servidor .....                                 | 25 |
| 3.3. Diseño de la aplicación .....                             | 26 |
| 3.3.1. Pantalla de inicio de sesión.....                       | 26 |
| 3.3.2. Pantalla del menú principal.....                        | 26 |
| 3.3.3. Pantalla del escáner de códigos QR .....                | 27 |
| 3.3.4. Pantalla de localización de una mesa .....              | 28 |
| 3.3.5. Pantalla de servicios.....                              | 29 |
| 3.3.6. Pantalla de búsqueda.....                               | 30 |
| 3.3.7. Pantalla de ayuda.....                                  | 30 |
| 3.4. Diseño del juego.....                                     | 31 |
| 3.5. Diseño de la base de datos.....                           | 33 |
| 3.6. Diseño conexión cliente-servidor.....                     | 34 |
| 3.7. Diseño de códigos QR .....                                | 35 |
| 4. Implementación .....                                        | 38 |
| 4.1. Medios utilizados .....                                   | 38 |
| 4.2. Implementación de la aplicación .....                     | 39 |
| 4.3. Implementación del servidor .....                         | 40 |
| 4.4. Implementación de la base de datos.....                   | 46 |
| 4.5. Implementación de conexión cliente-servidor.....          | 49 |
| 4.6. Implementación del escáner de códigos QR .....            | 52 |
| 4.7. Implementación de la conexión servidor-base de datos..... | 53 |
| 5. Resultados finales .....                                    | 57 |
| 6. Evaluación .....                                            | 64 |
| 7. Planificación y presupuesto.....                            | 68 |
| 7.1. Planificación .....                                       | 68 |
| 7.2. Presupuesto.....                                          | 72 |
| 8. Conclusiones y trabajo futuro .....                         | 74 |
| Bibliografía.....                                              | 76 |

|                                                         |    |
|---------------------------------------------------------|----|
| Summary .....                                           | 79 |
| Introduction.....                                       | 79 |
| State of art.....                                       | 80 |
| Design .....                                            | 83 |
| Implementation.....                                     | 85 |
| Evaluation .....                                        | 87 |
| Planification and budget.....                           | 87 |
| Conclusions and future work.....                        | 88 |
| Anexos .....                                            | 91 |
| Anexo 1: Pentagrama para el nivel 1. ....               | 91 |
| Anexo 2: Crucigrama para el nivel 5.....                | 91 |
| Anexo 3: Encuesta de satisfacción de la aplicación..... | 92 |

## Índice de Figuras

|                                                                                                  |    |
|--------------------------------------------------------------------------------------------------|----|
| Fig. 1: Entorno tecnológico del experimento de la UPF (Figura tomada de [3]) .....               | 6  |
| Fig. 2: Reacción general de los entrevistados hacia los códigos QR (Figura tomada de [4]) .....  | 7  |
| Fig. 3: Distribución global de smartphones en el 4º trimestre de 2012 (Figura tomada de [8]) ... | 9  |
| Fig. 4: Uso y calidad de las aplicaciones (Figura tomada de [8]) .....                           | 9  |
| Fig. 5: Vista del IDE Android Studio.....                                                        | 12 |
| Fig. 6: Visión del IDE NetBeans para Android.....                                                | 13 |
| Fig. 7: Evolución de popularidad de lenguajes de programación (Figura tomada de [19]) .....      | 14 |
| Fig. 8: Proceso de funcionamiento de Apache Tomcat .....                                         | 16 |
| Fig. 9: Interfaz de inicio de sesión de Google Sign-In.....                                      | 19 |
| Fig. 10: Interfaz de inicio de sesión de Facebook Login.....                                     | 20 |
| Fig. 11: Estructura de los códigos QR (Figura tomada de [32]) .....                              | 22 |
| Fig. 12: Diseño del servidor .....                                                               | 25 |
| Fig. 13: Boceto de la pantalla de inicio de sesión.....                                          | 26 |
| Fig. 14: Boceto de la pantalla del menú principal .....                                          | 27 |
| Fig. 15: Boceto de la pantalla del escáner de códigos QR .....                                   | 28 |
| Fig. 16: Ejemplo de imagen de localización de una zona de mesas de la planta 1 .....             | 28 |
| Fig. 17: Boceto de la pantalla de localización de una mesa.....                                  | 29 |
| Fig. 18: Boceto de la pantalla de servicios.....                                                 | 29 |
| Fig. 19: Boceto de la pantalla de búsqueda.....                                                  | 30 |
| Fig. 20: Boceto de la pantalla de ayuda .....                                                    | 31 |
| Fig. 21: Estructura de la base de datos.....                                                     | 33 |
| Fig. 22: Diagrama de inicio de sesión.....                                                       | 34 |
| Fig. 23: Ejemplo de conexión entre cliente y servidor.....                                       | 35 |
| Fig. 24: Diagrama de clases de los servlets.....                                                 | 42 |
| Fig. 25: Diagrama de clases de HttpServletRequest y HttpServletResponse.....                     | 43 |
| Fig. 26: Diagrama de flujo del servlet Login.....                                                | 44 |
| Fig. 27: Diagrama de flujo del servlet update.....                                               | 44 |
| Fig. 28: Diagrama de flujo del servlet restart .....                                             | 45 |
| Fig. 29: Diagrama de flujo del servlet nivel.....                                                | 45 |
| Fig. 30: Diagrama de flujo del servlet comprobarQR.....                                          | 46 |
| Fig. 31: Petición de objeto JSON con otro JSON adjunto.....                                      | 49 |
| Fig. 32: Petición de imagen al servidor.....                                                     | 52 |
| Fig. 33: Gráfico sobre la intención de uso diario de la aplicación .....                         | 64 |
| Fig. 34: Gráfico sobre la valoración de la experiencia global de uso.....                        | 65 |

|                                                                                          |    |
|------------------------------------------------------------------------------------------|----|
| Fig. 35: Fotografía nº. 1 de la prueba en la biblioteca .....                            | 65 |
| Fig. 36: Fotografía nº. 2 de la prueba en la biblioteca .....                            | 66 |
| Fig. 37: Fotografía nº. 3 de la prueba en la biblioteca .....                            | 66 |
| Fig. 38: Diagrama de Gantt.....                                                          | 71 |
| Fig. 39: General reaction of the interview to the QR codes (Figure taken from [4]) ..... | 81 |
| Fig. 40: Popularity of the programming languages (Figure taken from [19]) .....          | 82 |
| Fig. 41: Server's design .....                                                           | 83 |
| Fig. 42: Database's design .....                                                         | 84 |

## Índice de tablas

|                                                                                                 |    |
|-------------------------------------------------------------------------------------------------|----|
| Tabla 1: Diseño del juego .....                                                                 | 32 |
| Tabla 2: Lista de actividades del proyecto y explicación.....                                   | 40 |
| Tabla 3: Campos de la tabla Usuarios.....                                                       | 47 |
| Tabla 4: Campos de la tabla QRcodes.....                                                        | 48 |
| Tabla 5: Estado final de la tabla QRcodes de la base de datos.....                              | 49 |
| Tabla 6: Objetos JSON intercambiados para inicio de sesión válido .....                         | 50 |
| Tabla 7: Objetos JSON intercambiados para inicio de sesión no válido.....                       | 50 |
| Tabla 8: Objetos JSON intercambiados para comprobación de código QR existente.....              | 50 |
| Tabla 9: Objetos JSON intercambiados para comprobación de código QR no existente .....          | 50 |
| Tabla 10: Objetos JSON intercambiados para actualizar el nivel de un usuario existente .....    | 51 |
| Tabla 11: Objetos JSON intercambiados para actualizar el nivel de un usuario no existente ..... | 51 |
| Tabla 12: Objetos JSON intercambiados para reiniciar el nivel de un usuario existente.....      | 51 |
| Tabla 13: Objetos JSON intercambiados para reiniciar el nivel de un usuario no existente.....   | 52 |
| Tabla 14: Descripción y capturas de pantalla de las actividades de la aplicación.....           | 60 |
| Tabla 15: Capturas de pantalla de la localización de mesas de la biblioteca.....                | 61 |
| Tabla 16: Capturas de pantalla de las webs de información de las plantas de la biblioteca ..... | 63 |
| Tabla 17: Resultados de la prueba experimental.....                                             | 64 |
| Tabla 18: Planificación de tareas.....                                                          | 68 |
| Tabla 19: Presupuesto de mano de obra.....                                                      | 72 |
| Tabla 20: Fields of the table Usuarios.....                                                     | 87 |
| Tabla 21: Fields of the table QRcodes.....                                                      | 87 |
| Tabla 22: Cost of the manpower.....                                                             | 88 |





## 1. Introducción

En el presente capítulo de introducción se desarrollan los motivos, tanto personales como ajenos, que han conducido al diseño e implementación del proyecto. Asimismo, se exponen los objetivos que se pretenden conseguir, el marco regulador que influye en la aplicación desarrollada y la estructura seguida en esta memoria.

### 1.1. Motivación

La biblioteca de la universidad es un espacio en el que confluyen cada día cientos de usuarios muy variados, con distintos objetivos y con distintas necesidades. Un espacio tan diverso y con tantas necesidades diferentes que precisan ser atendidas necesita de una manera potente de informar acerca de sus recursos, sus posibilidades y su infraestructura.

Desde mi experiencia personal he aprendido lo difícil que puede llegar a ser, en especial durante los primeros días, ubicarse dentro de la biblioteca, llegar a conocer todos los servicios que ofrece, comprender para qué sirve cada espacio de la misma u obtener información de otra forma que no sea presencialmente.

Por lo comentado anteriormente y con el fin de hacerlo más liviano, el presente trabajo pretende ofrecer toda esta información de una forma divertida, creativa y atractiva: mediante un juego para la plataforma Android.

Este juego supone un desafío por niveles en el que el usuario debe utilizar la información acerca de la biblioteca para resolver distintos enigmas y responder preguntas. Además, buscando la total interacción con el espacio de la biblioteca, el juego se apoya en el uso de códigos QR repartidos a lo largo de la biblioteca y que el usuario debe escanear si quiere seguir avanzando en el juego.

### 1.2. Objetivos

El objetivo principal de este trabajo fin de grado ha sido el desarrollo de un juego para la plataforma Android que permita descubrir interactivamente todas las posibilidades de la biblioteca.

El juego se divide en distintos niveles que el alumno debe ir superando, poniendo a prueba su conocimiento sobre la biblioteca y descubriendo nuevos espacios a través de la búsqueda de pistas repartidas mediante códigos QR.

La experiencia del juego se ofrece de manera personalizada, teniendo en cuenta el desarrollo del usuario en el mismo. Por ello, la aplicación necesita de un método de autenticación de usuarios que permita personalizar el contenido de la aplicación según el usuario que se encuentra con la sesión iniciada.

Además, con el fin de guardar el avance de los usuarios en el juego, ofrecerles la posibilidad de reiniciar su progreso y llevar a cabo una operación de control sobre los usuarios la realización de este trabajo ha supuesto la creación de una base de datos y la implementación de un servidor web sencillo.

Así, podemos exponer los objetivos de manera resumida como sigue:

- Selección de un lenguaje de programación adecuado para la programación de la aplicación Android.
- Diseño e implementación de un juego para la plataforma Android basado en la biblioteca de la Universidad Carlos III de Madrid.
- Implementación de un sistema de autenticación de usuarios y control de acceso.
- Implementación de un sistema de localización de mesas de la biblioteca que ayude a los usuarios a ubicar a sus compañeros.
- Desarrollo de una base de datos para almacenar los datos de los usuarios , el nivel del juego en el que se encuentran, los códigos QR y las imágenes.
- Elección de una tecnología concreta de servidor web que permita realizar las funciones que son necesarias para el correcto funcionamiento de la aplicación.
- Implementación de un servidor web para la autenticación de usuarios, comprobación de códigos QR, suministro de imágenes y páginas web y almacenamiento de la información de los usuarios.

### 1.3. Marco regulador

La aplicación sobre la que trata este trabajo fin de grado maneja datos de usuarios para almacenarlos en una base de datos y ofrecer una experiencia personalizada. Por ello, debemos analizar el tratamiento que darle a estos datos según el RGPD (Reglamento General de Protección de Datos).

El RGPD [1] es un reglamento europeo obligatorio desde el 25 de mayo de 2018 que sustituye a la antigua LOPD española (Ley Orgánica de Protección de Datos) cuya función es regular la protección del uso de los datos personales de los ciudadanos de la Unión Europea. Su principal objetivo es otorgar a los ciudadanos más control sobre el uso que las empresas hacen de sus datos personales y favorecer el libre movimiento de datos en la Unión Europea.

La LOPD distinguía tres niveles de protección acumulativos para el tratamiento de datos personales, sin embargo el nuevo reglamento no establece niveles de seguridad. El RGPD [2] determina que las medidas de seguridad han de ser tales que garanticen una seguridad adecuada al riesgo, sin especificar niveles de seguridad ni medidas de cada uno de ellos.

En el caso del presente trabajo fin de grado, solo se manejan datos del usuario que incluyen nombre, apellidos y correo electrónico. Estos datos son datos muy básicos y por ello, el nivel de riesgo es relativamente bajo y requiere de pocas medidas de seguridad.

Estos datos citados anteriormente, son almacenados en un servidor seguro de la Universidad Carlos III de Madrid, garantizando el nivel de protección necesario acorde al riesgo que supone su uso malintencionado.

#### 1.4. Estructura de la memoria

Con el fin de exponer de una forma clara y ordenada el proceso completo hasta obtener la aplicación final, la memoria se ha estructurado de la siguiente manera:

1. **Introducción:** el presente capítulo, tal y como se ha comentado antes, pretende analizar las razones o motivaciones por las que se ha llevado a cabo el presente trabajo. A su vez, expone los objetivos perseguidos, el marco regulador que rodea al proyecto y la estructura de la memoria.
2. **Estado del Arte:** abarca el conocimiento de trabajos previos relacionados así como el estudio y comparación de las diferentes técnicas y tecnologías disponibles para alcanzar los objetivos ya expuestos. Concretamente, serán evaluadas las opciones de lenguajes de programación para aplicaciones Android, los distintos tipos de bases de datos y de servidores web.
3. **Diseño:** contiene las justificaciones de diseño previas a la creación de la aplicación Android, la base de datos, los códigos QR y el servidor web necesarios para el correcto funcionamiento del conjunto final.
4. **Implementación:** presenta el estudio de la implementación de los distintos elementos que conforman el sistema: aplicación, servidor, base de datos y las conexiones entre ellos.
5. **Resultados finales:** comprende la visualización del estado final de la aplicación objeto del presente trabajo.
6. **Planificación y presupuesto:** consiste en una descomposición en tareas del trabajo realizado, así como una estimación del tiempo que ha sido dedicado a cada una de ellas. Ofrece también un presupuesto estimado de realización del proyecto.
7. **Evaluación:** recopila los resultados experimentales recogidos durante una prueba presencial del sistema propuesto.
8. **Conclusiones y trabajo futuro:** destaca las conclusiones finales tras la implementación del proyecto y las posibles mejoras o añadidos a la aplicación en el futuro.



## 2. Estado del Arte

### 2.1. Introducción

Las aplicaciones para el sistema operativo Android se encuentran en una fase de crecimiento exponencial, lo que ha provocado una proliferación de nuevas tecnologías, entornos de desarrollo y opciones de programación.

En este capítulo de la memoria vamos a analizar trabajos previos relacionados con el proyecto así como las diferentes tecnologías existentes para la aplicación cliente-servidor y la base de datos que almacenará los datos básicos de sus usuarios.

### 2.2 Trabajos previos

#### 2.2.1. Universidad Pompeu Fabra

En el año 2010 en la Universidad Pompeu Fabra (Barcelona, España) [3] se llevó a cabo un experimento con alumnos de nuevo ingreso con el fin de explorar el campus de la universidad mediante tres medios diferentes. Los medios elegidos fueron una página web, un lector NFC instalado en un *smartphone* y, por último, una exploración visual.

La tarea que tenían los nuevos alumnos era conocer los distintos espacios del campus y obtener información sobre ellos y, a partir de esta información, contestar un cuestionario y elaborar una presentación. La información ofrecida por los tres medios era la misma, simplemente presentada de forma diferente.

Se emplearon 46 códigos NFC, que se encontraban repartidos por todo el campus, pegados a una cartulina amarilla para favorecer su visibilidad. Contenían imágenes, vídeos y audios explicando los distintos espacios del campus. Dichos contenidos también se encontraban disponibles en la página web y en carteles.

Uno de los resultados de la investigación fue que la tecnología NFC fue fácilmente adoptada por los alumnos y los profesores para este propósito. En este sentido, cabe esperar que una tecnología similar como la usada en este trabajo fin de grado, los códigos QR, también tendrán una fácil acogida en cuanto a dificultad de uso.

Además, el grupo que usó la tecnología móvil para explorar el campus obtuvo mejores resultados en el cuestionario y mostró una actitud más original en sus respuestas. Demostrando así que el uso de tecnologías móviles es una buena aproximación para la integración de actividades exploratorias formales e informales.

Finalmente, cabe destacar que los alumnos describieron la actividad con móviles como divertida, útil, interesante e innovadora y que fue un elemento motivador para el descubrimiento del campus.

La solución tecnológica llevada a cabo fue descrita por la universidad como un ejemplo de cómo las buenas prácticas educativas y la nueva tecnología pueden beneficiarse la una a la otra. Por ello, la universidad expresó su intención de seguir

desarrollando proyectos en contextos de aprendizaje con el uso de tecnologías como NFC O GPS para poder determinar la sostenibilidad del uso de estas técnicas.

En la Fig. 1 puede observarse el entorno tecnológico usado durante la investigación realizada en la Universidad Pompeu Fabra.

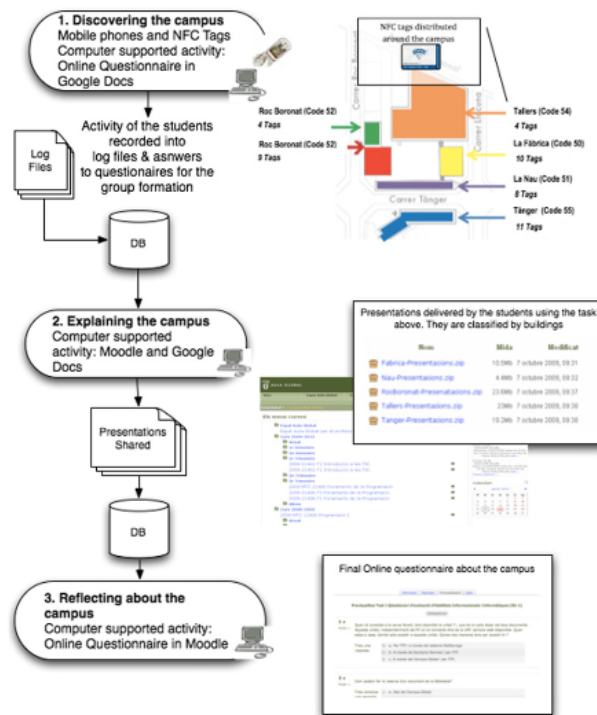


Fig. 1: Entorno tecnológico del experimento de la UPF (Figura tomada de [3])

Este artículo de investigación refuerza la idea del uso de elementos tecnológicos como los propuestos en este trabajo fin de grado para la exploración de la biblioteca, afirmando que aporta frescura, innovación y buenos resultados.

### 2.2.2. Universidad de Ryerson y Museo de Arte Inuit

En el año 2013 en Toronto (Canadá) se llevaron a cabo pruebas a sistemas con códigos QR previamente instalados en dos instituciones distintas: la Universidad de Ryerson y el Museo de Arte Inuit [4].

- La Universidad de Ryerson usaba códigos QR para distintos propósitos, entre los que se incluía un código a la entrada para acceder a una audioguía, códigos que redirigían a las páginas web de distintos servicios de la biblioteca e, incluso, códigos que mostraban información del catálogo de libros. El principal público de este servicio, al ser una universidad, eran los jóvenes universitarios.
- El Museo de Arte Inuit contaba con códigos QR al lado de cada pieza de arte para ofrecer más información sobre ella: un mapa del país de origen del autor, un vídeo del autor creando la obra, análisis del estilo, etc. Estos códigos instalados cerca de las obras expuestas permitían además dejar

comentarios sobre ellas. Finalmente, también se encontraban expuestos varios códigos QR en distintos puntos del museo que enlazaban con las redes sociales del mismo, facilitando así su seguimiento. El público de este sistema de códigos QR era el público general del museo, no un colectivo concreto.

Se llevaron a cabo entrevistas al público y a los trabajadores de ambas instituciones cinco veces a la semana durante un total de dos semanas. De este modo, se entrevistó a 12 trabajadores y 16 visitantes de cada institución, contando con un total de 56 participantes.

Los resultados determinaron que, pese a que la actitud de los entrevistados hacia los códigos QR fue mayoritariamente positiva (tal y como puede observarse en la Fig. 2), los entrevistados en la biblioteca mostraron más entusiasmo hacia los códigos que los entrevistados en el museo.

Además, la posesión de un teléfono capaz de leer códigos QR era más probable por parte de los universitarios y el 56% de ellos afirmaba haberlos utilizado alguna vez, frente al 25% de los visitantes del museo.

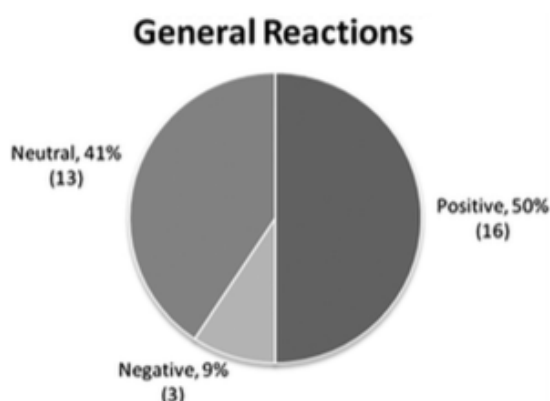


Fig. 2: Reacción general de los entrevistados hacia los códigos QR (Figura tomada de [4])

Sin embargo, el personal y los estudiantes de la biblioteca eran menos conscientes en cuanto al uso de códigos QR que los trabajadores del museo. Quizá porque el sistema del museo era un sistema formal, mientras que el de la biblioteca era experimental. Por ello, es importante la promoción del sistema implantado si se quiere conseguir una implementación exitosa.

### 2.2.3. Otros usos documentados

Robin Ashford, de la Universidad George Fox, relata en 2010 en el artículo *QR codes and Academic Libraries: reaching mobile users* [5] estar implementando su propio sistema de códigos QR en su biblioteca. Este sistema, afirma, cuenta con enlaces a videos introductorios sobre la biblioteca, a revistas y artículos electrónicos disponibles en la institución, video tráilers, DVD y más recursos.

La autora ofrece diversos ejemplos para los usos de códigos QR en bibliotecas, entre los que se incluyen:

- En estanterías con periódicos, artículos y revistas para enlazar con su correspondiente recurso electrónico.
- Enlaces a audio guías de la biblioteca con el fin de la orientación en su interior.
- Contacto con el personal de la biblioteca.
- En las salas de estudio, para proporcionar acceso a la página web de reserva de las mismas.
- Video tutoriales de la biblioteca.
- Información adicional sobre servicios e instalaciones.

## 2.3. Tecnologías en el lado del cliente

Este proyecto sigue el modelo cliente-servidor [6], un modelo donde las tareas se distribuyen entre los proveedores de recursos o servicios, llamados **servidores**, y los solicitantes de dichos recursos o servicios, llamados **clientes**. El cliente realiza peticiones de recursos o servicios al servidor y este le da respuesta.

El cliente es la parte fundamental de este trabajo, suponiendo la interfaz de interacción entre el usuario y el servidor. Esta interfaz se basa en una aplicación que el usuario podrá ejecutar en su móvil. Por ello, exploraremos las distintas alternativas que disponemos para dicho propósito.

### 2.3.1. Plataforma

Definimos la plataforma como el soporte, sistema operativo o recurso que permite la implementación de la aplicación que queremos diseñar. Consideramos para nuestro propósito tres plataformas distintas: los sistemas operativos móviles iOS y Android y la plataforma web. La elección de una plataforma condiciona el número de usuarios alcanzables, las tecnologías disponibles y los objetivos que pueden lograrse por lo que es importante analizar lo qué puede ofrecer cada una de ellas.

#### 2.3.1.1. iOS

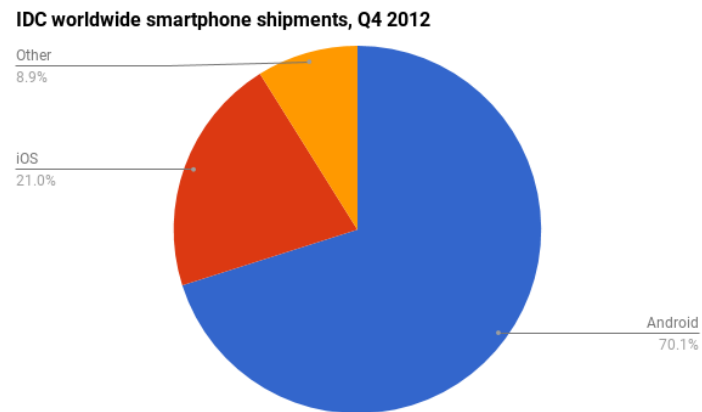
iOS (*iPhone Operating System*) [7] es el sistema operativo creado por Apple Inc. exclusivamente para sus dispositivos móviles. Fue lanzado en Junio de 2007 para el *smartphone* de la marca, el *iPhone*, pero actualmente se encuentra desplegado en otros dispositivos, tales como el *iPad*, *iPod* o *Apple Watch*.

La exclusividad de este sistema operativo supone una limitación en el número de usuarios finales, por lo que es conveniente estudiar las cuotas de mercado de este sistema operativo frente a otros.

Centrándonos únicamente en el mercado de los *smartphones*, en la Fig. 3 podemos observar que iOS consigue únicamente un 21% de la cuota de mercado mientras que el sistema operativo Android cuenta con un 70.1% según datos del último trimestre del



2012 [8]. Si bien estas cifras han cambiado a favor del sistema operativo de Apple en los últimos años, sigue existiendo una diferencia notable entre ambos sistemas que ha de ser considerada.

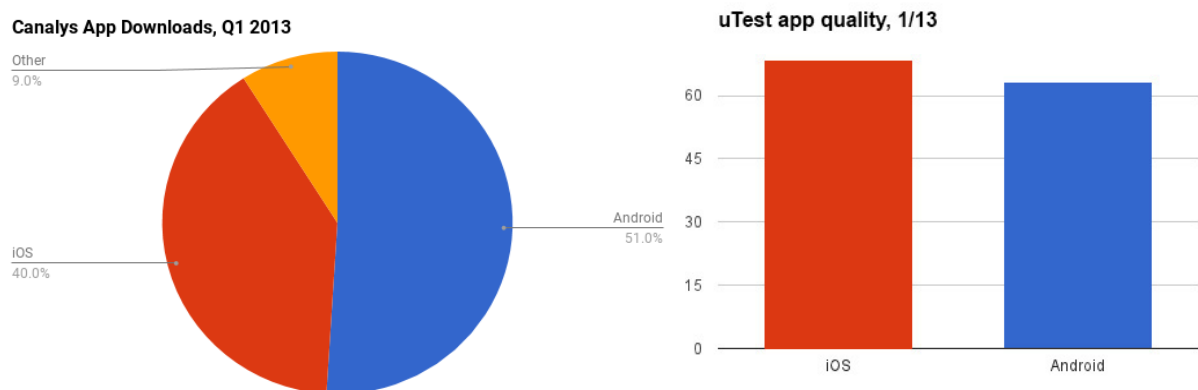


*Fig. 3: Distribución global de smartphones en el 4<sup>a</sup> trimestre de 2012 (Figura tomada de [8])*

Sin embargo, más ajustadas son las cifras de descarga de aplicaciones según plataforma: 51% de las aplicaciones son descargadas para la plataforma Android y el 40% para iOS.

Respecto a la calidad de las aplicaciones según opiniones de usuarios, iOS es el ganador con un 68,5% de satisfacción frente al 63,3% que alcanza Android.

En la Fig. 4 podemos observar las cifras de descarga de aplicaciones por plataforma y la calidad de las mismas.



*Fig. 4: Uso y calidad de las aplicaciones (Figura tomada de [8])*

En cuanto al lenguaje de programación, las aplicaciones en iOS se desarrollan en Swift. Este lenguaje de programación multiparadigma fuertemente tipado es propiedad de Apple y, al igual que el sistema operativo, es exclusivo para sus dispositivos.

La dificultad de aprender un nuevo lenguaje de programación se ve, en parte, equilibrada por el hecho de que Swift está formado en gran parte por elementos de C y Objective-C [9].

#### 2.3.1.2. Android

Android es el sistema operativo propiedad de Google LLC desde el año 2005, aunque no sería hasta 2007 cuando se lanzara la primera versión del sistema operativo Android. Actualmente, es el sistema operativo y plataforma de desarrollo más extendida globalmente.

Se trata de una plataforma de desarrollo libre basada en Linux y de código abierto, lo que significa que se puede usar y personalizar sin pagar *royalties* (pagos por derechos de autor).

Adicionalmente, Android [10] se adapta a cualquier tipo de hardware, ya sean móviles, tablets, relojes, coches o electrodomésticos. Esto supone una potencial ventaja, especialmente en un mundo cada vez más orientado al IoT (*Internet Of Things*), pero también un esfuerzo adicional para el programador.

Las aplicaciones son desarrolladas en Java, un lenguaje de programación popular en la actualidad y que ofrece grandes funcionalidades, lo que facilita su diseño e implementación. El principal elemento de una aplicación Android es la *activity*, que se define como una pantalla diseñada según una plantilla o *layout* y que se comporta de una determinada manera ante las interacciones del usuario mediante código Java.

#### 2.3.1.3. Web

Una aplicación web [11] es un tipo de software accesible a través de internet y desarrollado para ser soportado por navegadores web. Esta plataforma ofrece grandes ventajas respecto a las anteriores:

- Es independiente del sistema operativo: todo lo que se necesita para su ejecución es un navegador, sea cual sea el sistema operativo en uso.
- El cliente no posee carga operacional: las tareas de procesamiento se encuentran en el servidor por lo que el cliente es un mero medio de transporte entre éste y el usuario.
- La actualización de contenido es posible sin reinstalación.
- Requiere un menor tiempo de desarrollo.
- Posibilidad de uso de lenguajes de programación extensamente conocidos: HTML y CSS.

En contrapartida con las anteriores ventajas, el uso de esta plataforma restringe las posibilidades de programación y de uso del hardware del dispositivo. Además, la conexión a internet es necesaria incluso para aquellas funciones que, a priori, no precisarían de ella.

#### 2.3.1.4. Elección final

La plataforma elegida para el desarrollo de este proyecto ha sido la del sistema operativo Android. Las razones que han conducido a esta decisión se exponen a continuación:

- La alternativa de aplicación web ha sido descartada debido a la dificultad que supone implementar en HTML un juego como el propuesto haciendo uso de funciones de hardware como la cámara.
- En contraste con iOS, Android supone un mayor alcance de usuarios dentro de la Universidad Carlos III de Madrid por lo que resulta ser la opción más atractiva para el despliegue de la aplicación.
- Conocimientos previos del autor del lenguaje de programación Java, facilitando la tarea de documentación y preparación previa.
- Mayor facilidad para la distribución de la aplicación [12] en la plataforma Android, debido a la posibilidad de publicación fuera de la tienda *Google Play*.

#### 2.3.2. Entorno de desarrollo integrado

Un entorno de desarrollo integrado conocido por sus siglas en inglés IDE (*Integrated Development Environment*) es un programa o aplicación informática que ofrece distintos servicios integrados que facilitan la tarea de desarrollo de software del programador.

En especial, para el desarrollo de software para Android destacan entre los IDE más populares Android Studio, Eclipse y NetBeans.

##### 2.3.2.1. Android Studio

Android Studio [14] es el IDE oficial de Google para el desarrollo de aplicaciones para su plataforma Android. Por ello, toda la documentación oficial sobre la programación asociada a esta tarea hace referencia a este entorno de desarrollo.

Previo al lanzamiento de este entorno, Google apostaba por el uso de Eclipse, un IDE genérico para Java y otros lenguajes de programación. La falta de especialización en la plataforma de Android, la necesidad del uso de un *plugin* para ello y los buenos resultados de Android Studio en sus primeros años llevaron a Google a llamar a la migración de Eclipse a Android Studio por parte de los usuarios. En la actualidad, Google [13] ha declarado como obsoleto el *plugin* que permitía desarrollar aplicaciones en Eclipse y promueve el uso de Android Studio.

El uso de este IDE, al ser el oficial, facilita la tarea de seguir las reglas de diseño marcadas por Google y la incorporación de librerías a los proyectos, ahorrando tiempo y recursos durante el diseño.

A diferencia de su predecesor Eclipse, presenta una interfaz diseñada específicamente para la programación Android, optimizando el espacio disponible para ofrecer una vista completa del proyecto: estructura, estado, elementos, mensajes, errores, archivo actual, etc. En la Fig. 5 podemos observar dicha interfaz gráfica.

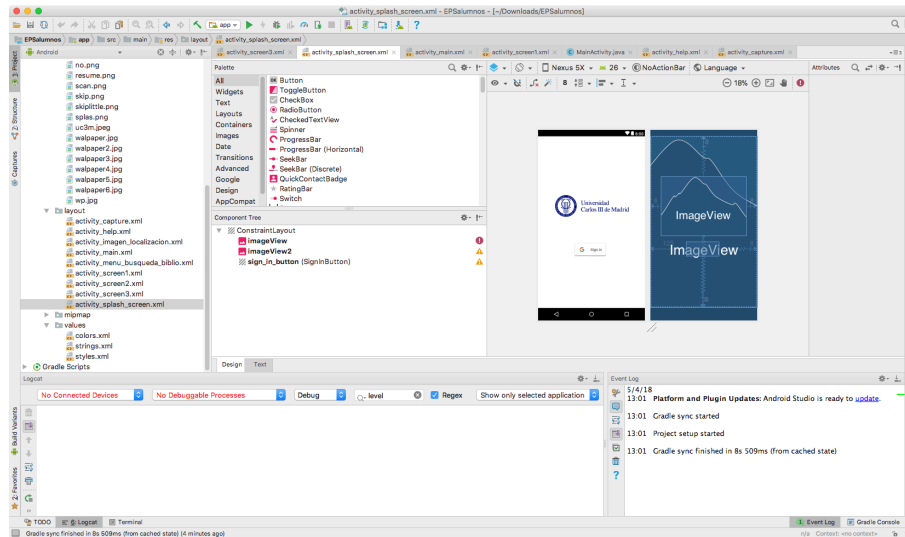


Fig. 5: Vista del IDE Android Studio

Entre alguna de sus características más notables destaca la posibilidad de ejecución instantánea en un terminal Android conectado al ordenador o en un terminal virtual instalado en la máquina desde esta misma herramienta. Además, su función *Instant Run* permite realizar cambios en el código mientras la app se ejecuta, sin tener que volver a compilar y ejecutar la aplicación.

Por último, cabe mencionar que *Gradle* [15] es implementado en todos los proyectos de Android Studio. Se trata de una herramienta para la automatización de la construcción del código de los proyectos. Entre las ventajas que proporciona su uso, se encuentran las siguientes:

- Reutilización de código de forma sencilla.
- Sencillez para configuración y personalización de la compilación.
- Gestiona las dependencias de forma potente y cómoda.
- Facilidad para el diseño de diferentes versiones de la aplicación (móvil/tablet, gratuito/pago...).

### 2.3.2.2. NetBeans

NetBeans IDE [16] es un entorno de desarrollo gratuito y de código abierto sin restricciones de uso iniciado por Sun Microsystems. Es usado para desarrollar, principalmente, aplicaciones Java para escritorio, móviles y webs.

Cuenta con todas las funcionalidades propias de un IDE para Java: edición inteligente de código, organización inteligente del proyecto, interfaz de usuario rápida y detector de errores.

Permite la personalización para el desarrollo de aplicaciones para el sistema operativo de Google mediante la instalación del plugin *NBAndroid* [17]. Este plugin incorpora al IDE genérico las siguientes funcionalidades:

- Soporte para el kit de desarrollo de software de Android (Android SDK).
- Incorporación de LogCat para la documentación de errores, advertencias y mensajes de log.
- Editores XML mejorados para adaptarse a los archivos XML de Android.
- Vista previa de la interfaz gráfica de usuario, tal y como puede observarse en la Fig. 6.
- Construcción de proyecto mediante Gradle.
- Posibles extensiones para incorporar nuevas funciones.

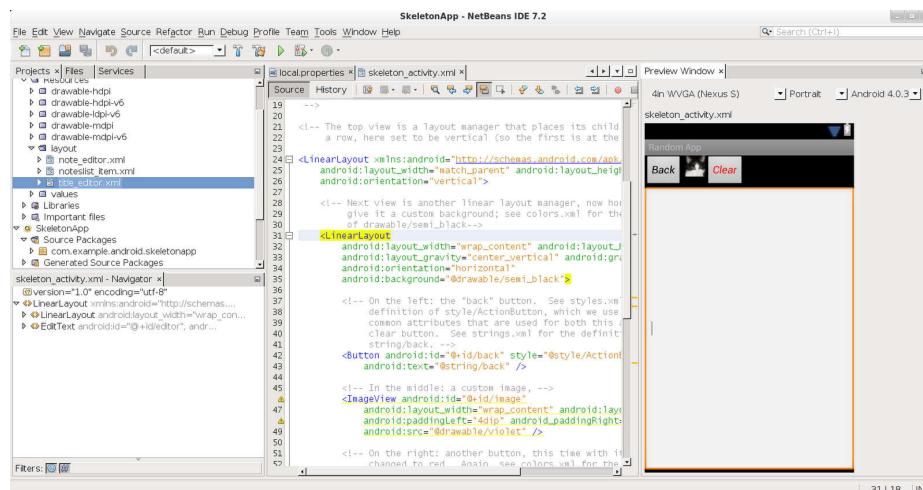


Fig. 6: Visión del IDE NetBeans para Android<sup>1</sup>

### 2.3.2.3. Elección final

El entorno de desarrollo integrado escogido finalmente fue Android Studio frente a NetBeans. Ambos entornos proporcionaban herramientas análogas e interfaces gráficas realmente similares. Sin embargo, la disponibilidad de toda la documentación

<sup>1</sup> <http://www.i-programmer.info/images/stories/News/2012/OCT/A/netbeansandroidview.jpg>

sobre desarrollo de aplicaciones orientada a Android Studio supuso el criterio decisivo para ser el elegido.

Otros aspectos considerados fueron un menor uso de memoria RAM por parte de Android Studio, una instalación más sencilla y una mayor facilidad de integración con otros productos de Google usados en nuestro proyecto, como Google Sign-In.

## 2.4. Tecnologías en el lado del servidor

Siguiendo la arquitectura cliente-servidor, el servidor es el encargado de recibir las peticiones del cliente y responderle, suministrándole los servicios y recursos requeridos.

El servidor que precisa este proyecto es un servidor simple, que permita la autenticación de usuarios, devuelva al cliente los datos que precise sobre su progreso en el juego y que guarde y extraiga datos de una base de datos.

Por tanto, teniendo en cuenta los bajos requerimientos exigidos al servidor de nuestro proyecto, analizamos a continuación las tecnologías existentes para la programación del mismo.

### 2.4.1. Lenguajes de programación

Un lenguaje de programación [18] es una forma organizada de comunicarse con un ordenador, haciendo que se comporte acorde a las instrucciones introducidas por el programador. Está constituido por una serie de herramientas que permiten crear un software o programa con un objetivo.

Existe una gran cantidad de lenguajes de programación, con características y orientaciones diferentes. Por ello, es necesario analizar las estadísticas de uso de los lenguajes de programación para determinar cuál es más conveniente emplear para este trabajo fin de grado.

Como se puede observar en la Fig. 7, según el ranking TIOBE en el intervalo entre el año 2002 y 2018 los lenguajes más populares fueron Java, C, C++ y Python [19]

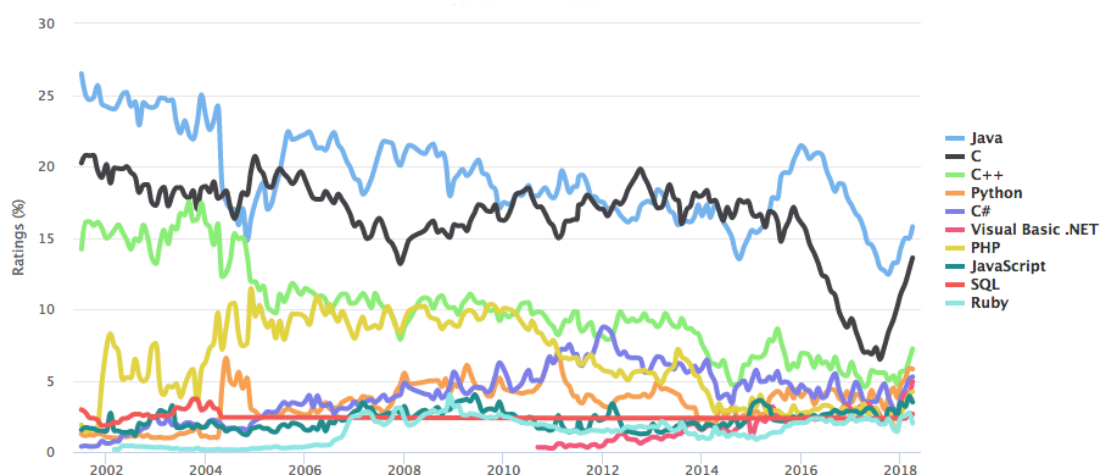


Fig. 7: Evolución de popularidad de lenguajes de programación (Figura tomada de [19])

Debido a que C y C++ no son frecuentemente utilizados en el desarrollo web, vamos a centrarnos en analizar los dos restantes: Python y Java.

#### 2.4.1.1. Python

Python [18] es un lenguaje de programación de código abierto desde su creación en 1991. Se trata de un lenguaje de propósito general multiparadigma con soporte para programación orientada a objetos, procedimental, modular y algunos aspectos de programación funcional.

Posee características avanzadas de procesamiento de cadenas de texto e interfaces para operar con comandos del sistema, sockets y programación web.

La principal herramienta asociada al desarrollo web en Python es Django [20], un entorno de trabajo gratuito y de código abierto con gran potencial para desarrollar sitios web y servidores.

Sin embargo, pese al gran potencial de Django y su amplia documentación en la red, es una herramienta que precisa una gran cantidad de tiempo para aprender a explotar sus posibilidades.

#### 2.4.1.2. Java

Java [18] es un lenguaje de programación fuertemente tipado y principalmente orientado a objetos. Fue publicado en 1995 por Sun Microsystems con la intención de que una vez escrito el código pudiera ser ejecutado en cualquier dispositivo (WORA: *Write Once, Run Anywhere*). Esto se consigue gracias a la implementación de una máquina virtual (*Java Virtual Machine*) para evitar las consideraciones sobre el hardware del dispositivo que ejecuta el código.

Su sintaxis deriva en gran medida de C y C++ y se trata del lenguaje de programación más popular en el mundo debido a su gran portabilidad, eficiencia y sencillez.

No se entiende hablar de programación web de servidores en Java sin hablar de *servlets*. Los *Java Servlets* son módulos de código Java que se utilizan en un servidor web para ampliar sus capacidades de respuesta a los clientes usando las funcionalidades del lenguaje. La herramienta de mayor uso con *servlets* es Apache Tomcat, un contenedor de *servlets* que permite el desarrollo de un servidor web de manera simple.

#### 2.4.1.3. Elección final

Finalmente, el lenguaje de programación elegido fue Java. Las razones que llevaron a su elección fueron un conocimiento previo del lenguaje por parte del autor y una mayor simplicidad de las herramientas asociadas a dicho lenguaje, en especial del uso de *servlets* y de Apache Tomcat.

### 2.4.2. Apache Tomcat

Apache Tomcat [21] es un herramienta software de código abierto que funciona como un contenedor de *servlets* de Java. Su función principal como servidor web es recibir las peticiones del cliente y direccionarlas a los *servlets*.

Está desarrollado en Java por lo que funciona en cualquier sistema operativo que posea máquina virtual Java. Simplifica el proceso de comunicación entre el cliente y el servidor, siendo necesario únicamente programar los distintos *servlets* y configurar adecuadamente la herramienta Tomcat.

El proceso de funcionamiento de Tomcat puede verse resumido en la Fig. 8.

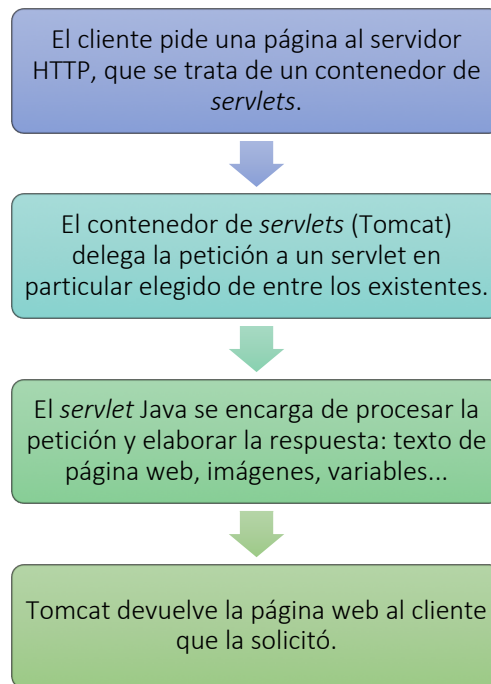


Fig. 8: Proceso de funcionamiento de Apache Tomcat

Debido a su sencillez y a su gran potencial fue la única tecnología considerada a la hora de realización de un servidor web en Java. La versión utilizada fue Apache Tomcat 8.0.28

### 2.5 Base de datos

La base de datos que vamos a desarrollar es muy sencilla, por lo que usaremos un SGDB (sistema de gestión de base de datos) de tipo relacional.

Una base de datos relacional [22] almacena los datos en tablas, así como las relaciones entre ellos. Cada fila de la tabla tiene un identificador único, llamado clave principal y una serie de columnas que almacena un determinado tipo de datos con un valor real, relacionados todos con la misma entidad u objeto.

Podemos elegir entre dos tipos de SGDB, atendiendo al modo de distribución [23]:



- **SGBD comerciales:** sistemas bajo una licencia comercial con código cerrado, lo que supone restricciones en su uso o modificación. A pesar de existir opciones gratuitas, en la mayoría de los casos es necesario pagar una cuota por su uso.
- **SGBD libres:** sistemas bajo una licencia libre, lo que supone la libertad absoluta para su uso, estudio, modificación y distribución. Aunque la gran mayoría son gratuitos, es conveniente separar la asociación “libre=gratuito”, pues existen opciones por las que hay que pagar los costes de distribución.

Tras valorar las distintas opciones, se ha optado por el uso de SGBD libres debido a su total libertad de uso y con fin de evitar incidencias con la licencia durante la implementación.

Una vez elegido el tipo de SGBD que vamos a usar, valoraremos algunas de las opciones que ofrecen un SGBD libre de tipo relacional. Para ello es necesario hablar de SQL, usado por la inmensa mayoría de SGBD relacionales.

SQL [24] es un lenguaje de programación estandarizado usado para cambiar datos y objetos de bases de datos mediante consultas (en inglés, *queries*). Es un lenguaje declarativo en el que se describe lo que se quiere obtener pero no como obtenerlo (a diferencia de los lenguajes procedimentales como C o Java).

Las versiones más simples de SQL y, por tanto, las que han sido analizadas son: MySQL, SQLite y PostgreSQL.

- **MySQL:** es una implementación sencilla [25] enfocada principalmente en aplicaciones web de lectura, donde lo más importante es la optimización de consultas sencillas. A pesar de su sencillez, es la base de datos de sitios de tal popularidad como Facebook o Youtube debido a su gran rapidez, fluidez y facilidad de uso.
- **SQLite:** es la implementación más sencilla [26]. No tiene una definición estricta de tipos de datos y su arquitectura completa se basa en un único fichero estructurado de datos.
- **PostgreSQL:** su principal ventaja [27] es la alta concurrencia: mientras un proceso escribe en la tabla, otros pueden acceder a la misma tabla sin bloqueos. Además, a pesar de no ser un lenguaje puramente orientado a objetos, incluye características de la orientación a objetos: herencias, tipos de datos, funciones, etc.

Teniendo en cuenta las ventajas de los tres lenguajes de programación, el escogido ha sido **MySQL** por tres razones:

- Gran orientación hacia aplicaciones web.

- El único usuario que accede a la base de datos es el servidor, por lo que la alta concurrencia no es necesaria en este proyecto, descartando PostgreSQL.
- Otra razón para elegir MySQL frente a SQL es una mejor y más clara definición de los tipos.

## 2.6. Comunicación Cliente-Servidor

El cliente y el servidor necesitan estar en continuo contacto para intercambiar la información que precisen en cada momento. Por ello, es necesario analizar las tecnologías usadas para lograr una comunicación fluida entre nuestro cliente y nuestro servidor.

### 2.6.1. Volley

Volley [28] es una librería HTTP que simplifica y agiliza las peticiones y respuestas web de las aplicaciones Android. Se trata de una herramienta de código abierto disponible en la plataforma *Github* y su uso se encuentra documentado en la página oficial de desarrolladores de la plataforma Android.

Entre las características y funcionalidades que ofrece, se citan las siguientes:

- Planificación automática de las solicitudes de red.
- Múltiples conexiones concurrentes.
- Soporte para prioridad de solicitudes.
- Posibilidad de cancelación de solicitudes.
- Facilidad de personalización.

El uso de esta librería al programar una aplicación con Android Studio es muy sencillo, contando con métodos ya implementados para solicitar un objeto del tipo JSON, una imagen o una cadena de texto y esperar a su recepción.

### 2.6.2. JSON

El formato de datos JavaScript Object Notation (JSON) [29] es un derivado de los literales del lenguaje de programación JavaScript y es conocido como el estándar para el intercambio de datos entre clientes y servidores.

Se trata de una representación textual de información definida por un pequeño conjunto de reglas con las que los datos son estructurados. La especificación define que los datos pueden ser estructurados de dos formas distintas:

1. Una colección de pares nombre/valor.
2. Una lista ordenada de valores.

Es un formato interoperable, conseguido mediante una gramática simple que puede ser traducida a cualquier lenguaje simplemente entendiendo la gramática.

Usaremos este formato de datos para intercambiar información entre cliente y servidor debido a que se trata del formato de uso más extendido y a su facilidad de integración en el lenguaje Java.

### 2.6.3. Inicio de sesión

El inicio de sesión supone una forma de autenticar a un usuario en concreto de cara al servidor, es una definición inequívoca del usuario que quiere conectar con el servidor. También supone una medida de seguridad, evitando posibles suplantaciones y usos maliciosos.

La existencia en el proyecto de múltiples usuarios que se comunican con el servidor para obtener una experiencia personalizada hace necesaria la implementación de algún tipo de inicio de sesión en el servidor.

Dicho inicio de sesión puede hacerse con cuentas ya existentes mediante el uso de APIs de servicios de terceros como Google o Facebook o mediante métodos de autenticación y de sesión propios.

#### 2.6.3.1. Google Sign-In

Google Sign-In permite un proceso de autenticación simple mediante el uso de una cuenta de cualquier servicio de Google. Proporciona los elementos visuales como el botón de inicio de sesión y las distintas ventanas, así como una extensa biblioteca de funciones para configurar el servicio.

Permite además, procesar la autenticación en un servidor *backend* [30] mediante el uso de un *token* (una cadena alfanumérica única y aleatoria) enviado al servidor una vez realizado satisfactoriamente el inicio de sesión. De esta forma, el servidor puede obtener los datos del usuario que ha iniciado sesión (correo, nombre, apellidos...) sin necesidad de que estos viajen por la red.

El uso de este servicio con Android Studio se encuentra ampliamente documentado, contando con una guía detallada de su implementación sobre dicho IDE. La interfaz de inicio de sesión que proporciona Google Sign-In puede observarse en la Fig. 9.

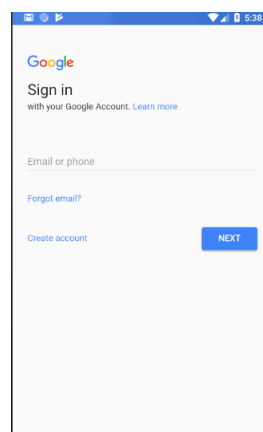


Fig. 9: Interfaz de inicio de sesión de Google Sign-In

### 2.6.3.2. Facebook Login

Facebook Login [31] , al igual que el servicio análogo de Google, simplifica el inicio de sesión de usuarios mediante una biblioteca de funciones, elementos gráficos y clases ya creadas, usando esta vez una cuenta del servicio de Facebook.

Si bien es posible el uso de un servidor *backend* para procesar la autenticación, su implementación no es tan simple ni se encuentra tan bien detallada como el uso de Google Sign-in, dificultando la tarea del programador.

En cuanto al resto de funciones, sus funcionalidades y operaciones son similares y producen el resultado deseado: identificación del usuario. La interfaz gráfica de inicio de sesión puede observarse en la Fig. 10.

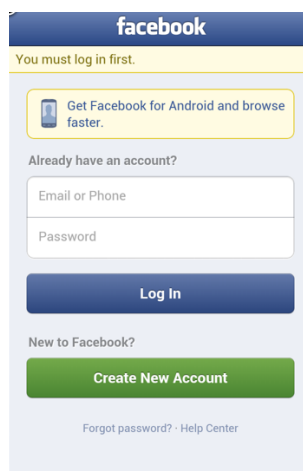


Fig. 10: Interfaz de inicio de sesión de Facebook Login

### 2.6.3.3. Métodos propios

La implementación de un inicio de sesión puede realizarse de forma totalmente autónoma, sin la dependencia de ningún servicio de terceros. Esto supone una total autonomía al definir el perfil del usuario y los datos necesarios para su creación, el nivel de confidencialidad de los datos, así como en el control y definición de la sesión.

Esta gran ventaja, la autonomía, se ve contrapuesta a los grandes requisitos técnicos que supone su implementación:

- Necesidad de cifrado propio: es necesario dotar al sistema de un cifrado para los datos de los usuarios con el fin de evitar ataques de suplantación y otros usos indebidos. En el caso de usar servidores de terceros era su responsabilidad pero ahora recae sobre nosotros.
- Base de datos más elaborada para almacenar los datos de inicio de sesión de los usuarios (correo electrónico y contraseña).
- Servidor más complejo: es ahora nuestro servidor el encargado de cotejar que el usuario y contraseña introducidos coinciden con los registrados en la base de datos de usuarios.

- Acuerdo de confidencialidad: el administrador del proyecto tiene acceso a los datos de inicio de sesión de los usuarios, lo que exige un acuerdo de confidencialidad por su parte.
- Implementación más compleja: diseño de la interfaz de conexión y otros elementos como el botón de inicio de sesión y administración de la comunicación cliente-servidor para el inicio de sesión.

#### 2.6.3.4. Elección final

Debido a los grandes retos que exigía la implementación manual de un servicio de inicio de sesión propio, dicha opción fue descartada y se decidió usar un sistema de terceros. Además, es conocido que los usuarios suelen ser reacios a la creación de una cuenta nueva y prefieren usar una cuenta ya existente con el fin de ahorrar tiempo.

El uso de un servicio de terceros supone, como ya se ha expuesto, una pérdida del control y de la autonomía. En contraposición, cuentan con sus propias bases de datos, cuya gestión es su responsabilidad y proporcionan la interfaz gráfica de inicio de sesión, los métodos de cifrado y la confidencialidad requerida.

Entre los servicios proporcionados por Google y Facebook, finalmente fueron los primeros los elegidos: Google Sign-In. La razón principal para esta elección es la constancia de que todos los usuarios potenciales de la aplicación poseen una cuenta de Google proporcionada por la propia universidad. Sin embargo, no podemos afirmar lo mismo de la posesión de una cuenta de Facebook. Este hecho hace imposible la utilización exclusiva del servicio de Facebook pues el público alcanzado sería mucho menor. Además, la utilización del servicio de Google parecía más sencilla y estaba mejor documentada que su análogo de Facebook.

### 2.7. Códigos QR

Un código QR (*Quick Response*) [32] es un código de barras bidimensional que puede almacenar datos codificados. Permiten la codificación de distintos recursos: texto, páginas web, tarjetas de contacto, geolocalizaciones, descarga de archivos, imágenes...

Es el sucesor del código de barras tradicional en una dimensión y, como ventaja frente a este, puede almacenar más información en un espacio menor. También incorpora una mayor recuperación frente a errores que su predecesor unidimensional.

Fue inventado en 1994 por la compañía japonesa Denso Wave con el principal objetivo de agilizar las lecturas de los códigos de barras, por eso lo denominaron *Quick Response Codes* (códigos de respuesta rápida, en inglés). Al contrario que otros códigos bidimensionales, es un proyecto de código abierto donde los derechos de patente de la empresa creadora no son ejercidos. Esto ha permitido una explosión de aplicaciones, mejoras y progreso en el uso de estos códigos.

Se caracterizan por la existencia de tres cuadrados en las esquinas superiores izquierda y derecha e inferior izquierda, que permiten la localización del código por parte del lector. La estructura general de un código QR es presentada en la Fig. 11.



Fig. 11: Estructura de los códigos QR (Figura tomada de [32])

Una de sus principales características es que puede ser leído en cualquier orientación gracias a los elementos de posicionamiento y alineación. Además, la información máxima que puede almacenar un código QR puede ser aumentada mediante la introducción de módulos.

Según el número de módulos que incorpora el código QR y, por tanto, de la información capaz de almacenar, se definen diferentes versiones de códigos QR: desde la versión 1 (con 21x21 módulos) a la versión 40 (177x177 módulos).

Otra forma de aumentar la capacidad del código QR es disminuir la capacidad de corrección. Para dicho propósito, se definen cuatro niveles de corrección:

- Nivel L (bajo): 7% del código se puede restaurar.
- Nivel M (medio): 15% del código se puede restaurar.
- Nivel Q (calidad): 25% del código se puede restaurar.
- Nivel H (alto): 30% del código se puede restaurar.

Sin embargo, la clasificación más interesante para estos códigos es según si su contenido puede cambiar en el tiempo o no:

- Códigos QR **estáticos**: codifican la información una vez que se generan. Una vez establecido el contenido no es posible cambiarlo.
- Códigos QR **dinámicos**: son reciclables, permiten cambiar el contenido tantas veces como se desee. Además, proporcionan estadísticas de uso.

La extrema sencillez para crear estos códigos y la incorporación de lectores de códigos QR a los *smartphones* han supuesto una verdadera revolución, pasando de estar orientados a servicios de empresa (como stock) a comunicación con el **cliente**. Por ello, los códigos QR han permitido la interacción con el entorno a través del teléfono móvil y es común encontrarlos actualmente en revistas, paradas de autobús, libros e incluso, entornos físicos.



### 3. Diseño

En este capítulo abordaremos el diseño de la aplicación, la base de datos y del servidor. En primer lugar expondremos los requisitos que han de cumplir ambos y posteriormente nos centraremos en el diseño específico y la estructura de ambos.

#### 3.1. Requisitos

##### 3.1.1. Requisitos del servidor

1. Comunicación con el cliente para el intercambio de datos.
2. Obtención de datos de inicio de sesión mediante el uso de servicios de Google.
3. Almacenamiento y actualización de datos de inicio de sesión en la base de datos.
4. Distribución de imágenes al cliente bajo demanda.
5. Actualización en la base de datos del nivel de juego actual del cliente.
6. Comunicación al cliente de su nivel actual en el juego.
7. Reinicio del nivel de juego bajo petición del cliente.
8. Comprobación de recursos asociados a un código QR leído por el cliente.
9. Devolución de recursos asociados a un código QR.

##### 3.1.2. Requisitos del cliente

10. Pantalla de inicio de sesión
11. Pop-up de error en caso de inicio de sesión incorrecto o servidor no disponible.
12. Autenticación con servicios de Google.
13. Envío de *token* de autenticación al servidor.
14. Petición de nivel actual de juego al servidor.
15. Pantalla con el menú principal mostrando todas las opciones de la aplicación.
16. Botón para cerrar sesión.
17. Implementación de un escáner de códigos QR.
18. Pantalla de ayuda para explicar todas las funcionalidades.
19. Pantalla con enlaces a distintos servicios de la biblioteca.
20. Pantalla para búsqueda de un código de localización dentro de la biblioteca.
21. Petición al servidor de reinicio de nivel al pulsar la opción “Nuevo juego”.
22. Petición de confirmación al usuario para reiniciar el nivel.
23. Envío de peticiones de actualización de nivel tras superar cada nivel.
24. Continuación del juego a partir del nivel actual.
25. Pantallas para las distintas actividades del juego.
26. Obtención de imágenes del servidor.
27. Petición al servidor de recursos asociados a un código QR leído.
28. Pantalla que muestre la localización de una mesa y permita compartirla.

##### 3.1.3. Requisitos de la base de datos

28. Tabla para almacenar datos de usuarios.
29. Tabla para almacenar códigos QR con su recurso asociado.



30. Distinción entre imagen, enlace u otro tipo de recurso.

### 3.2. Diseño del servidor

La tarea del servidor es atender las peticiones del cliente, realizar operaciones y/o comprobaciones para, finalmente, devolver una respuesta que informe del resultado al cliente.

En concreto, en este proyecto, el servidor debe validar con Google y guardar información sobre el inicio de sesión de los usuarios, modificar el nivel en el que se encuentran y devolver los recursos asociados a los códigos QR:

- Inicio de sesión: el servidor obtiene los datos del usuario que ha iniciado sesión usando los servicios de Google mediante la validación de un *token* enviado por el cliente. A continuación, almacena o actualiza dichos datos (nombre, email, fecha de inicio de sesión) en la base de datos.
- Modificación de nivel de usuario: permite la modificación, actualización y reinicio del nivel del juego en el que se encuentra un determinado usuario.
- Devolución de recursos de códigos QR: comprobación de que el código QR escaneado por el cliente se corresponde con algún recurso existente en la base de datos. En caso afirmativo, devuelve al cliente dicho recurso (página web, texto, imagen...).

Así, el servidor debe operar con el cliente, las bases de datos y los servicios de Google tal y como se describe en la Fig. 12.

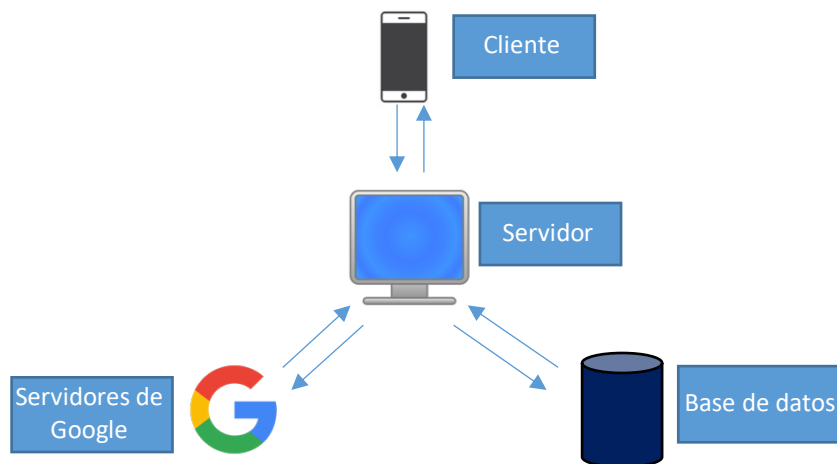


Fig. 12: Diseño del servidor

Más adelante expondremos en detalle el esquema de conexiones del servidor con el cliente y los servidores de Google.

### 3.3. Diseño de la aplicación

Para cumplir con todos los requisitos y funcionalidades del cliente, lo más importante en cuanto a diseño de la aplicación, es hacer un esbozo de las diferentes pantallas y clarificar donde se va a realizar cada una de las funcionalidades.

Con este propósito, distinguimos a continuación las siguientes pantallas:

#### 3.3.1. Pantalla de inicio de sesión

Se trata de la primera pantalla que verá el usuario al abrir la aplicación.

Principalmente debe contar con un botón que una vez pulsado enlace con el servicio de inicio de sesión con Google. Para manejar posibles desconexiones del servidor, podemos incluir un *pop-up* que informe de la incidencia y permita al usuario salir de la aplicación.

En la Fig. 13 podemos ver una maqueta de cómo podría ser esta primera pantalla de inicio de sesión.



Fig. 13: Boceto de la pantalla de inicio de sesión

#### 3.3.2. Pantalla del menú principal

El usuario accederá a ella una vez ha iniciado sesión correctamente. En esta pantalla se deben ofrecer al usuario todas las especificaciones anteriormente descritas. Por ello, organizamos todas las funcionalidades en seis categorías:

- **Nuevo juego:** para iniciar el juego de pistas por la biblioteca desde el principio, desde el nivel 1.
- **Continuar:** para volver al juego en el nivel en el que se quedó la última vez.
- **Escanear:** para escanear los distintos códigos QR repartidos por la biblioteca (códigos para localización de mesas, de información, de servicios, etc.)

- **Servicios:** para ofrecer un enlace directo a distintos servicios sin tener que escanear su código QR correspondiente (reserva de aulas, horario de la biblioteca, préstamo interbibliotecario...)
- **Buscar:** para conseguir la imagen asociada a la localización de una mesa usando la etiqueta que acompaña al código QR.
- **Ayuda:** para explicar el funcionamiento de las cinco categorías anteriores.

Estas seis categorías se ofrecerán en forma de cuadrícula con un botón circular y texto debajo con el nombre de la categoría. Los botones serán imágenes que intentarán reflejar la funcionalidad asociada y, una vez pulsados, encaminarán a la pantalla que corresponda.

Además, esta pantalla del menú principal debe contar con un botón para cerrar sesión que permita al usuario volver a la pantalla de inicio de sesión y poder cambiar de cuenta.

Finalmente, para evitar posibles errores añadiremos un *pop-up* de confirmación al pulsar el botón “Nuevo juego”, pidiendo la interacción del usuario para llevar a cabo la acción.

En la Fig. 14 observamos un boceto de lo descrito anteriormente.

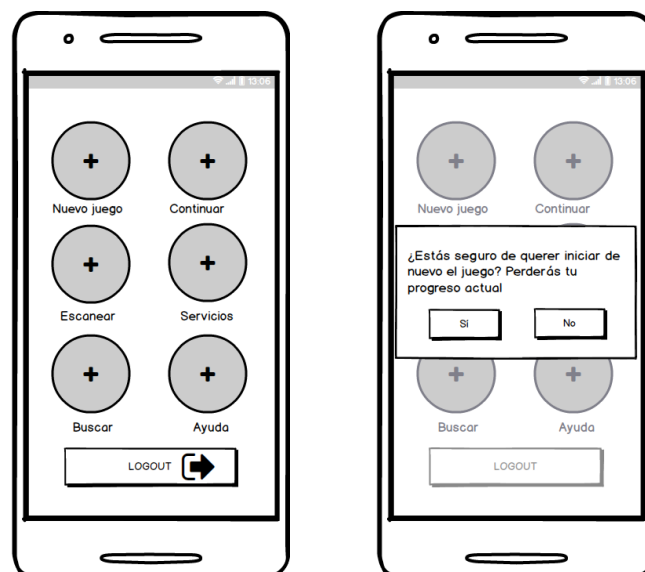
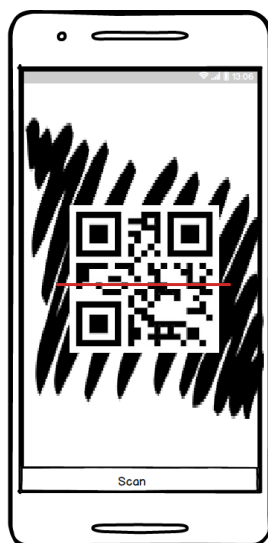


Fig. 14: Boceto de la pantalla del menú principal

### 3.3.3. Pantalla del escáner de códigos QR

Esta pantalla no ha sido diseñada por completo, sino que se basa en el diseño ya ofrecido por la herramienta usada para escanear los códigos QR. Dicho diseño se basa en una pantalla que muestra la cámara trasera del usuario, un cuadrado donde encajar el código QR a escanear y una pequeña barra roja para centrar el código. Como opción adicional, se puede agregar en la parte inferior un mensaje aclaratorio o una instrucción.

En la Fig. 15 podemos observar con detalle un boceto de la pantalla anteriormente descrita.

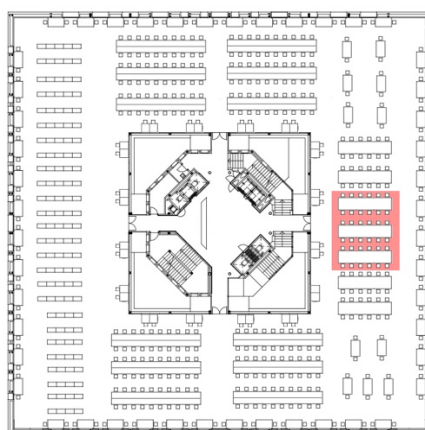


*Fig. 15: Boceto de la pantalla del escáner de códigos QR*

#### 3.3.4. Pantalla de localización de una mesa

Esta pantalla es accedida al escanear uno de los códigos QR repartidos por las mesas de la biblioteca para poder compartir la ubicación de los usuarios de la aplicación. También puede ser accedida mediante la función “Buscar”, como explicaremos más adelante.

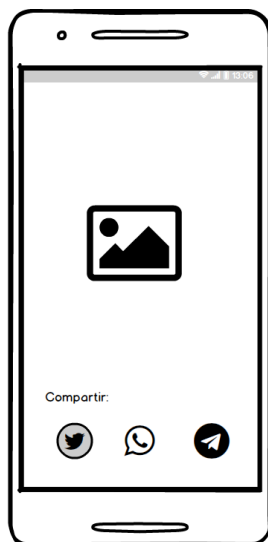
Su función es mostrar una imagen: el plano de la planta de la biblioteca correspondiente con la localización de la mesa señalada. Para ello, se han editado los planos de las mesas de la biblioteca y dividido en zonas (ejemplo en la Fig. 16). Además, para facilitar su uso, será posible hacer zoom desplazando dos dedos en dirección contraria.



*Fig. 16: Ejemplo de imagen de localización de una zona de mesas de la planta 1*

Finalmente, para poder cumplir con su objetivo y facilitar el uso al usuario evitando tener que realizar capturas de pantalla, se incorporan tres botones para compartir la imagen mediante tres redes sociales: Twitter, Whatsapp y Telegram.

En la Fig. 17 puede verse un boceto de la pantalla de localización de una mesa.

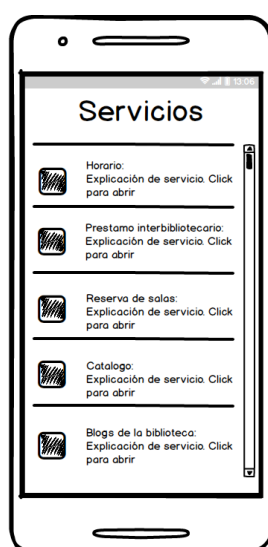


*Fig. 17: Boceto de la pantalla de localización de una mesa*

### 3.3.5. Pantalla de servicios

Esta pantalla muestra a modo de lista desplazable todos los recursos a los que se puede acceder directamente sin necesidad de escanear el código QR correspondiente (a pesar de que también es posible acceder a través de ellos).

Cada recurso se presenta como un botón formado por un pequeño icono representativo y una descripción del recurso que se va a obtener al presionarlo tal y como puede observarse en la Fig. 18.



*Fig. 18: Boceto de la pantalla de servicios*

### 3.3.6. Pantalla de búsqueda

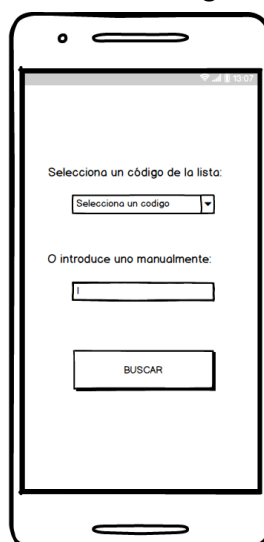
Es posible acceder a las imágenes de localización de las mesas sin tener que escanear los códigos QR que se encuentran pegados en ellas, sin más que utilizar la etiqueta asociada a dichos códigos. Estas etiquetas acompañarán a los códigos QR diseñados para este propósito.

Para conseguir esto diseñamos esta pantalla que permite acceder a una imagen concreta de dos formas distintas:

- Seleccionando la etiqueta en una lista desplegable de posibles etiquetas.
- Introduciendo manualmente dicha etiqueta y pulsando un botón para buscar.

En ambos casos, la aplicación obtendrá la imagen del servidor de nuestro proyecto y la mostrará siguiendo el diseño descrito en el apartado 3.4.

El boceto de dicha pantalla se muestra en la Fig. 19.



*Fig. 19: Boceto de la pantalla de búsqueda*

### 3.3.7. Pantalla de ayuda

Se ha diseñado esta pantalla con el fin de ofrecer al usuario una pequeña explicación de cada categoría anteriormente descrita en esta memoria. El diseño elegido es muy similar al usado en la pantalla de servicios: una lista desplazable con la explicación de cada una de las categorías presentes en el menú principal y el icono asociado.

A diferencia de la pantalla de servicios, esta pantalla no tiene ningún enlace asociado al pulsar una de las opciones: es simplemente informativa. El boceto se muestra en la Fig. 20.



Fig. 20: Boceto de la pantalla de ayuda

### 3.4. Diseño del juego

Para diseñar un juego que sea de utilidad para los usuarios es importante determinar los niveles en los que se dividirá, la descripción y ubicación dentro de la biblioteca de cada uno de ellos y el objetivo asociado a dicho nivel. En la Tabla 1 puede observarse el diseño llevado a cabo para el juego de exploración en la biblioteca. Dicho juego se centra en evitar una intrusión ficticia en los servidores de la universidad. El usuario debe ayudar a detener dicha intrusión recopilando las instrucciones que un profesor redactó y repartió por la biblioteca.

| Nivel | Lugar                    | Descripción                                                                                                                                                                                                                                                                                                                                                                                  | Objetivo                                                                                      |
|-------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| 0     | Entrada a la planta baja | Introducción al juego y explicación de la mecánica.                                                                                                                                                                                                                                                                                                                                          | Toma de contacto con el juego y su objetivo.                                                  |
| 1     | Planta baja              | Se muestra una imagen con un texto escrito en modo espejo, donde se indica al usuario que pregunte a los bibliotecarios para pasar al siguiente nivel. Estos le indican que debe agitar el dispositivo para pasar al siguiente nivel y le entregan un pentagrama (véase Anexo I) y tres comodines de respuesta para el nivel 7. Se manda al usuario a la puerta de la Sala de Audiovisuales. | Exploración de la planta 0 de la biblioteca e interacción con los empleados de la biblioteca. |
| 2     | Planta -1                | Los usuarios se encontrarán con varios códigos QR sin ningún tipo de descripción. Escaneándolos deben descubrir que estos códigos reproducen las notas musicales en                                                                                                                                                                                                                          | Descubrimiento de la ubicación de la Sala de Audiovisuales y                                  |

|                |                               |                                                                                                                                                                                                                                                                                                                               |                                                                                                                      |
|----------------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
|                |                               | sus dispositivos. Para continuar deben reproducir la melodía del pentagrama.                                                                                                                                                                                                                                                  | exploración de la planta -1.                                                                                         |
| <b>3</b>       | Planta 1                      | Se indica al usuario que la siguiente pista está en una mesa de la planta 1. La ubicación de la mesa se muestra usando el sistema de localización desarrollado para este trabajo. Para pasar al siguiente nivel los usuarios deben resolver un enigma en binario que les devuelve la contraseña.                              | Evaluación del sistema de localización de mesas y exploración general de la planta 1.                                |
| <b>Nivel 4</b> | Planta 1                      | En pantalla se muestra sin más descripción el título y autor de una obra. Los jugadores deben usar el catálogo por medio de la aplicación o en los puestos informáticos de la biblioteca para localizar la pista que se encontrará en el estante de dicha obra.                                                               | Utilización del catálogo de la biblioteca y ubicación de una obra concreta.                                          |
| <b>Nivel 5</b> | Salas de grupo-<br>Planta 3   | Una instrucción mandará a los jugadores a las salas de grupo. Allí encontrarán un crucigrama sencillo sobre la biblioteca que deben resolver si quieren continuar (véase Anexo 2)                                                                                                                                             | Descubrimiento de las salas de grupo de la biblioteca y aprendizaje sobre los recursos y servicios de la biblioteca. |
| <b>Nivel 6</b> | Cualquier lugar               | Se mostrarán 9 preguntas sobre la biblioteca que el usuario debe responder. Puede usar los tres comodines que le fueron entregados en el nivel 1 para preguntar a los empleados de la biblioteca.                                                                                                                             | Aprendizaje sobre recursos y servicios de la biblioteca. Interacción con los empleados de la biblioteca.             |
| <b>Nivel 7</b> | Aulas informáticas – Planta 3 | El paso final para detener la intrusión en el sistema es introducir una clave en un ordenador de las aulas informáticas de la planta 3. El jugador debe encontrar dicho ordenador e introducir una clave. La clave es de 9 dígitos y el jugador podrá ver tantas letras de ella como preguntas acertara en el nivel anterior. | Exploración de las aulas informáticas y fin del juego.                                                               |

*Tabla 1: Diseño del juego*



### 3.5. Diseño de la base de datos

Es necesario listar todos los datos que queremos almacenar en la base de datos para poder organizarlos en tablas y establecer las relaciones entre los distintos campos. En nuestro caso, distinguimos dos propósitos claros para la base de datos: almacenar información de los usuarios y asociar códigos QR con recursos. Por ello, tiene sentido la creación de dos tablas totalmente diferenciadas y sin relación alguna: **Usuarios** Y **QRcodes**. Los datos a almacenar en cada una de ellas son:

#### Tabla Usuarios:

- **Nombre:** contiene el nombre de los usuarios que han iniciado sesión.
- **Apellido:** contiene el apellido de los usuarios que han iniciado sesión.
- **Email:** correo electrónico de los usuarios. Será la clave primaria de esta base de datos (es el campo que identifica unívocamente a la fila).
- **Último acceso:** día del último inicio de sesión en la aplicación de cada usuario.
- **Nivel:** campo que refleja el nivel del juego en el que se encuentra cada usuario.
- **Token:** almacena el último token de autenticación válido de un determinado usuario.
- **TokenHora:** guarda el momento en el que se recibió el último token válido de autenticación.

#### Tabla QRcodes:

- **Código:** etiqueta que identifica a cada código QR. Clave primaria de la tabla.
- **Recurso:** texto, enlace, imagen u otro recurso asociado a dicho código QR.
- **Tipo:** valor que indica el tipo de recurso del campo anterior (0 para imágenes, 1 para enlaces, 2 para páginas HTML internas al servidor)

Recopilando todo, la base de datos a la que accederá el servidor puede ser organizada tal y como se observa en la Fig. 21.

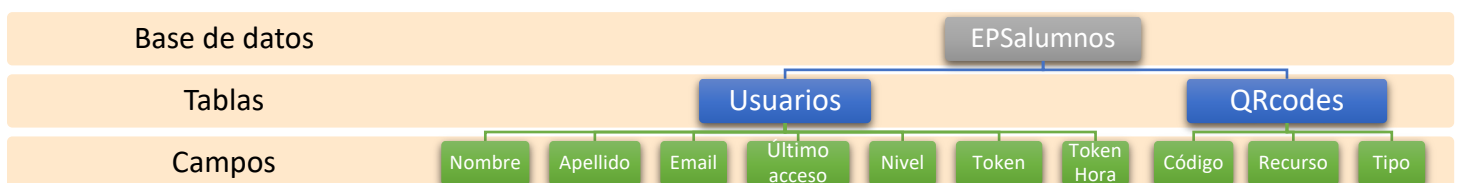


Fig. 21: Estructura de la base de datos

### 3.6. Diseño conexión cliente-servidor

La comunicación entre el cliente y el servidor es esencial para que todo funcione correctamente. Por ello, antes de implementar realmente dicha comunicación es importante diseñar el orden y forma en que va a producirse.

El primer paso para la comunicación es el inicio de sesión. En este caso cliente y servidor tienen un intermediario: el servicio de autenticación de Google. Los pasos del inicio de sesión, suponiendo éxito en la autenticación, pueden observarse en la Fig. 22 y son los siguientes:

1. Envío de usuario y contraseña al servicio de autenticación de Google desde la aplicación.
2. Respuesta de éxito desde Google incluyendo un *token* de autenticación válido durante 30 minutos.
3. Envío del *token* desde cliente a servidor.
4. Petición de datos de sesión a Google por parte del servidor a través del *token*.
5. Respuesta de Google con los datos de usuario. Almacenamiento de los datos de usuario y el token de autenticación en la base de datos del servidor
6. Respuesta de servidor a cliente para informar del éxito de la solicitud de conexión.

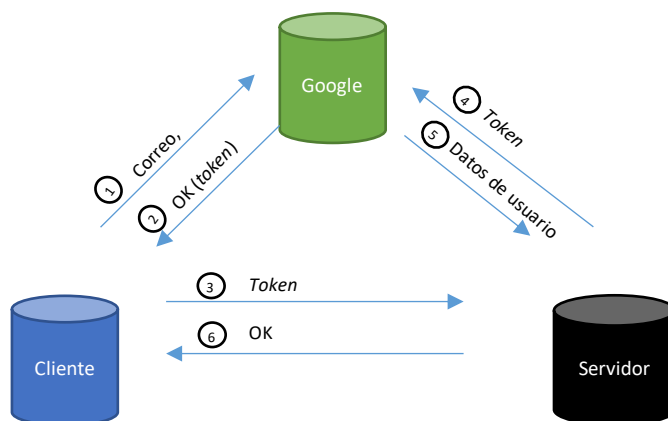


Fig. 22: Diagrama de inicio de sesión

A partir de aquí, según las acciones que haga el usuario así serán las respuestas del servidor. Por ejemplo, si un usuario pide comprobar un código QR, enviando para ello la lectura de dicho código, obtendrá del servidor el recurso asociado al mismo. Si lo que se solicita es una imagen, el servidor devolverá el fichero con dicha imagen. Si solicita continuar el juego, el servidor indicará a la aplicación el nivel en el que se encuentra el usuario.

Cabe destacar que todas las peticiones y respuestas usan HTTP, un protocolo no orientado a conexión por lo que es necesario acompañar las peticiones con un campo

que identifique al usuario, como el email. También debido a esto no es necesario finalizar la conexión entre cliente y servidor.

Además, para evitar ataques malintencionados todas las peticiones posteriores al inicio de sesión que supongan un cambio en la base de datos deben ir acompañadas del token de autenticación. Dicho token es válido durante 30 minutos a partir de la recepción pero se refresca al recibir una nueva petición válida con el mismo token. Sin embargo, para conseguir realmente la certeza de que el intercambio de mensajes es seguro sería necesario el uso de HTTPS (*Hypertext Transport Protocol Secure*). En el presente trabajo no ha sido implementado debido a que precisa de la existencia de un nombre de dominio y el pago de un certificado a una autoridad de certificación.

A modo de ejemplo, en la Fig. 23 se muestra el desarrollo normal de una conexión entre cliente y servidor mediante un uso sencillo de la aplicación: inicio de sesión y comprobación de un código QR.

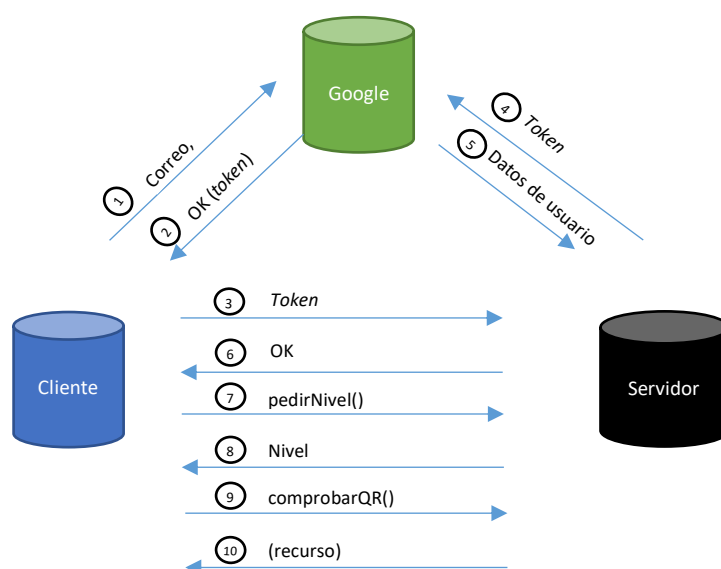


Fig. 23: Ejemplo de conexión entre cliente y servidor

### 3.7. Diseño de códigos QR

En este proyecto hemos optado por el uso de códigos QR estáticos debido a su gran sencillez de generación, mantenimiento y aplicación. Sin embargo, para permitir cambios en los contenidos, es el servidor el que asocia el código QR con el recurso que queremos devolver al usuario. Por ello, cambiando al recurso asociado a un código QR concreto en la base de datos del servidor conseguimos un comportamiento distinto al escanear dicho código con la aplicación en el cliente. En contrapartida, estos códigos QR solo tienen sentido con el uso de la aplicación diseñada para este propósito y no con ninguna otra.

Los códigos QR usados en este proyecto serán, por tanto, simples etiquetas asociadas a un recurso en la base de datos. Por ello, algunos códigos, como los usados para la localización dentro de la biblioteca, se acompañarán de su correspondiente etiqueta para facilitar la búsqueda de recursos por parte del usuario.

Los usos de los códigos QR usados en el sistema son:

- **Localización de mesas:** códigos QR que al ser escaneados devuelven la imagen de localización de la mesa en la que están.
- **Acceso a recursos:** códigos QR que permitirán acceder a recursos como el calendario de apertura, el catálogo, el préstamo interbibliotecario, la web de reserva de salas de grupo y otros servicios.
- **Información general de plantas:** un código QR en cada planta mostrará al ser escaneado la información sobre dicha planta y un plano explicado de la misma.



## 4. Implementación

Una vez diseñado en conjunto el servidor, la aplicación, la base de datos y las distintas conexiones entre ellos puede comenzar la implementación de los mismos. En el presente capítulo se expondrá el proceso de creación de los distintos elementos que conforman este trabajo.

### 4.1. Medios utilizados

A continuación se van a detallar los equipos tecnológicos y programas informáticos utilizados para la realización del presente TFG.

#### **Equipos:**

- Ordenador de sobremesa: iMac, finales de 2013.
  - Procesador Intel Core i5, 2.7 GHz
  - Memoria 8GB 1600 MHz DDR3
  - Gráficos Intel Iris Pro 1536 MB
  - Disco duro de 1 TB
  - Sistema Operativo: macOS High Sierra
- Ordenador portátil MacBook Air, principios de 2015.
  - Procesador Intel Core i5, 1.6 GHz
  - Memoria 4GB 1600 MHz DDR3
  - Gráficos Intel HD Graphics 6000 1536 MB
  - Disco duro de 128 GB
  - Sistema Operativo: macOS High Sierra

#### **Programas:**

- Android Studio 3.1 para macOS: Entorno de desarrollo utilizado para la creación de la aplicación Android, el cliente.
- VirtualBox 5.0.4: Máquina virtual que emula el sistema operativo de Linux, en concreto una distribución Ubuntu. Los requisitos de dicha máquina virtual se detallan a continuación:
  - Sistema Operativo: Ubuntu 16.04 (64 bits)
  - Memoria base: 2232 MB
  - Memoria de vídeo: 22 MB
  - Tamaño virtual: 9 GB

En concreto, en este sistema operativo emulado de Linux, los programas utilizados son los siguientes:

- Eclipse 3.8: Utilizado para la programación y compilación Java de los distintos servlets del servidor.
- Apache Tomcat 8.0.28: Contenedor de servlets utilizado para el lanzamiento del servidor.

- MySQL 5.7.22: gestión de la base de datos en el lado del servidor.
- Microsoft Word 2016 para macOS: editor de texto utilizado para la redacción de la presente memoria.

La aplicación fue desarrollada y ejecutada en el ordenador de sobremesa anteriormente descrito, mientras que el servidor se llevó a cabo en el ordenador portátil. La existencia de ambas unidades en distintos equipos permite probar el funcionamiento de la aplicación a través de la red local.

## 4.2. Implementación de la aplicación

Cada *activity* definida en Android Studio lleva asociada una clase Java y un objeto de estilo con extensión *xml*. En el archivo Java es donde se lleva a cabo la programación funcional, mientras que en el archivo de estilo se definen los elementos visuales, sus propiedades y distribución.

Con el fin de adaptar la aplicación al mayor número posible de usuarios, se han definido todas las pantallas usando plantillas que se ajustan a la resolución de pantalla de los dispositivos.

En la Tabla 2 puede observarse un listado de todas las *activities* que componen el proyecto de Android Studio, la función que realizan, los archivos asociados y el servlet del servidor con el que se comunican (en caso de obtención de imágenes, se denotará como media)

| Activity           | Archivos asociados                                          | Función                                                                                                                                                                | Servlet                         |
|--------------------|-------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| SplashScreen       | SplashScreen.java<br>Activity_splash_screen.xml             | Mostrar pantalla de inicio de sesión y gestionar la comunicación con el servidor para la gestión de dicha operación.                                                   | Login                           |
| MainActivity       | MainActivity.java<br>Activity_main.xml                      | Mostrar pantalla de menú principal y gestionar los resultados del escáner de códigos QR, la petición del nivel actual del usuario y la petición de reinicio del mismo. | Nivel<br>ComprobarQR<br>Restart |
| ServiciosScreen    | ServiciosScreen.java<br>Activity_servicios_screen.xml       | Mostrar la lista de recursos de la biblioteca y proporcionar el acceso a los mismos al hacer click sobre ellos.                                                        | -                               |
| ImagenLocalizacion | ImagenLocalizacion.java<br>Activity_imagen_localizacion.xml | Mostrar la imagen de localización de una mesa determinada de la biblioteca. Permite hacer zoom en la imagen y compartirla en redes sociales.                           | Media                           |

|                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                 |        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|--------|
| HelpActivity                                                                                                                                                             | HelpActivity.java<br>Activity_help.xml                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Mostrar la descripción de cada una de las opciones del menú principal                                                           | -      |
| MenuBusqueda                                                                                                                                                             | MenuBusqueda.java<br>activity_menu_busqueda.xml                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Mostrar el buscador de etiquetas de códigos QR de las mesas para acceder a las imágenes de localización sin escanear el código. | -      |
| CaptureActivity                                                                                                                                                          | CaptureActivity.java<br>Activity_capture.xml                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Permitir la apertura del escáner de códigos QR                                                                                  | -      |
| Screen1<br>Screen2<br>Screen3                                                                                                                                            | Screen1.java<br>Activity_screen1.xml<br>Screen2.java<br>Activity_screen2.xml<br>Screen3.java<br>Activity_screen3.xml                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Mostrar el nivel 0, la introducción al juego. Envío de petición de actualización de nivel.                                      | Update |
| Nivel1_0<br>Nivel1_1<br>Nivel2_0<br>Nivel2_1<br>Nivel2_2<br>Nivel3_0<br>Nivel3_1<br>Nivel4_0<br>Nivel5_0<br>Nivel5_1<br>intropreguntas<br>preguntasScreen<br>finalScreen | Nivel1_0.java<br>Nivel1_1.java<br>Nivel2_0.java<br>Nivel2_1.java<br>Nivel2_2.java<br>Nivel3_0.java<br>Nivel3_1.java<br>Nivel4_0.java<br>Nivel5_0.java<br>Nivel5_1.java<br>Intropreguntas.java<br>preguntasScreen.java<br>finalScreen.java<br>activity_nivel1_0.xml<br>activity_nivel1_1.xml<br>activity_nivel2_0.xml<br>activity_nivel2_1.xml<br>activity_nivel2_2.xml<br>activity_nivel3_0.xml<br>activity_nivel3_1.xml<br>activity_nivel4_0.xml<br>activity_nivel5_0.xml<br>activity_nivel5_1.xml<br>activity_intro_preguntas.xml<br>activity_preguntas_screen.xml<br>activity_final_screen.xml | Mostrar las distintas pantallas del juego y gestionar el avance a las siguientes.                                               | Update |

*Tabla 2: Lista de actividades del proyecto y explicación*

### 4.3. Implementación del servidor

Para cumplir con las especificaciones de diseño ya descritas el servidor ha sido estructurado de la siguiente forma dentro de la carpeta del servidor Tomcat:

- **Ruta webapps/comandos:** contiene los *servlets* desarrollados para cumplir los requisitos de diseño. Los archivos de Java ya compilados (*.class*) se encuentran



dentro de la ruta **WEB-INF/classes**, mientras que las librerías necesarias se encuentran en **WEB-INF/lib**.

Los *servlets* son accedidos mediante la URL: <http://direcciónIPdelServidor:8080/comandos/nombreServlet>. Cada uno de los *servlets* responde a una de las funcionalidades requeridas en el servidor y son los siguientes:

- Servlet **login**: encargado de comprobar el token de autenticación de los usuarios que inician sesión, actualizar la base de datos con los de inicio de sesión y responder al usuario con el resultado.
  - Servlet **update**: su función es recibir peticiones del cliente para la actualización del nivel de un usuario determinado en la base de datos.. Debe verificar la validez del token de autenticación, comprobar si ha expirado, actualizar el nivel del usuario en la base de datos y ofrecer una respuesta al cliente.
  - Servlet **restart**: recibe las peticiones del cliente para el reinicio del nivel de un determinado usuario en la base de datos.. Su funcionamiento es análogo al *servlet update* pero el nivel a actualizar siempre es 0.
  - Servlet **nivel**: devuelve al cliente el nivel del juego en el que se encuentra un usuario determinado a través de una consulta en la base de datos.
  - Servlet **comprobarQR**: recibe un código leído por el cliente y debe comprobar en la base de datos su existencia. Tras realizar esta consulta, devuelve al usuario el recurso y el tipo de recurso que tiene asociado ese código.
- Ruta **webapps/media**: contiene las imágenes que deben ser alojadas en el servidor. Se accederá a ellas mediante la siguiente URL: <http://direcciónIPdelServidor:8080/media/nombreImagen>. No es necesario programar nada para esta función, Tomcat sirve automáticamente las imágenes.
  - Ruta **webapps/web**: contiene los archivos HTML servidos directamente a la aplicación. De forma análoga se accede a ellos mediante la URL: <http://direcciónIPdelServidor:8080/web/nombreArchivo> y tampoco es necesario realizar ninguna acción, Tomcat los sirve de forma automática.

Con el fin de plasmar de forma visual la implementación de los *servlets*, en la Fig. 24 se muestra el diagrama de clases de los mismos.

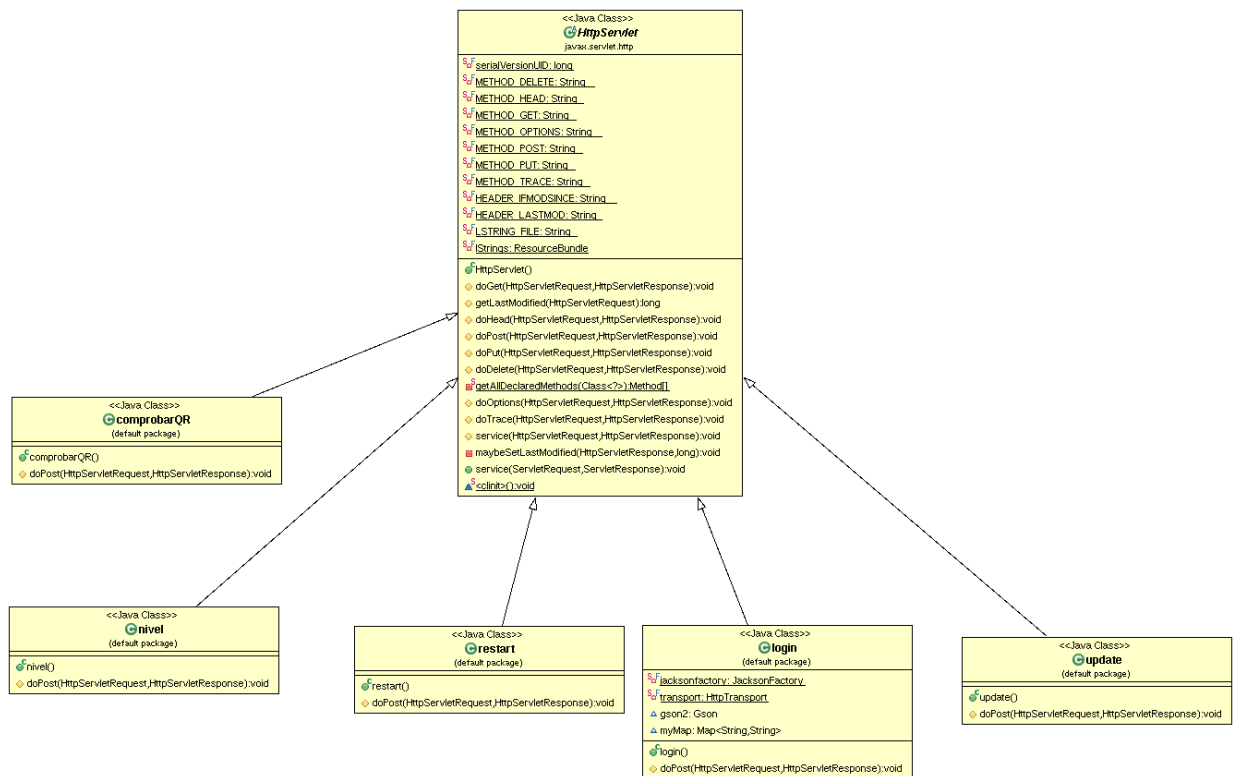


Fig. 24: Diagrama de clases de los servlets

Tal y como se puede observar en la Fig. 24, todos los *servlets* heredan directamente de la clase Java *HttpServlet* e implementan el método `doPost()`, que se ejecuta al recibir una solicitud HTTP del tipo POST. Este método recibe como parámetros dos objetos: *HttpServletRequest* y *HttpServletResponse* cuyo diagrama de clases se presenta en la Fig. 25 para ver los métodos y atributos que poseen.

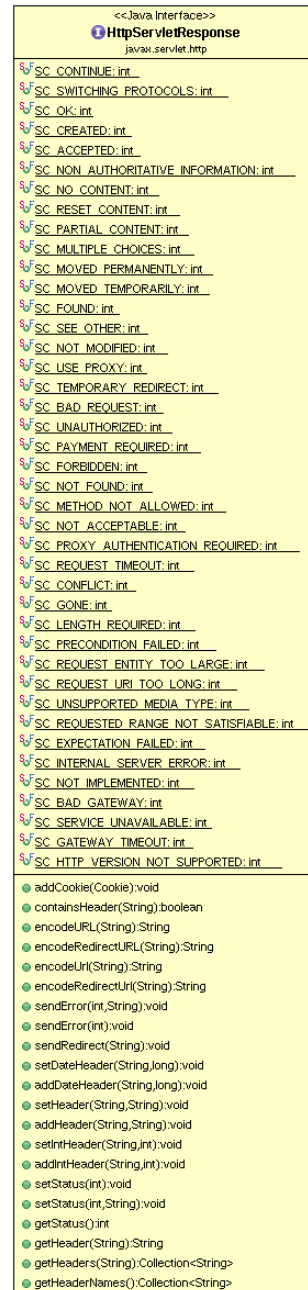
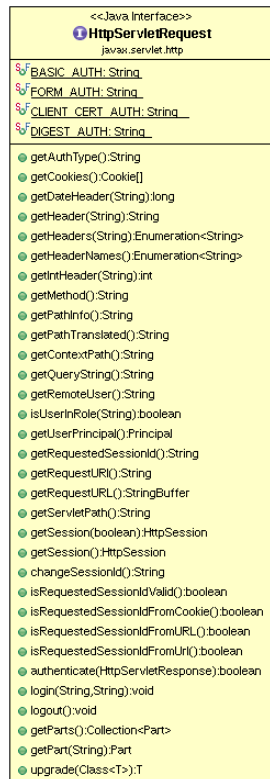


Fig. 25: Diagrama de clases de `HttpServletRequest` y `HttpServletResponse`

Finalmente, en las Fig. 26 a la 30 pueden observarse el diagrama de flujo de cada uno de los *servlets* que conforman el servidor para entender su funcionamiento interno.

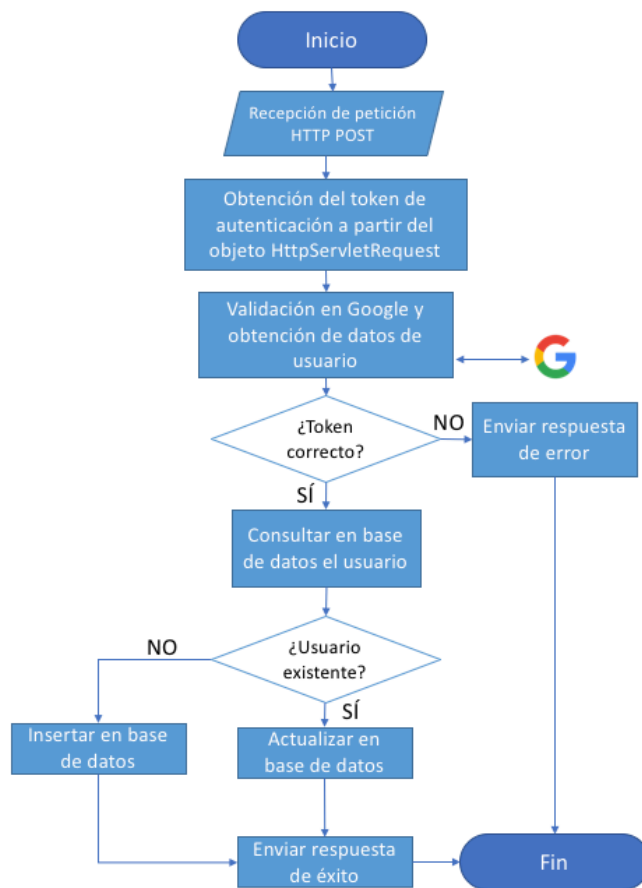


Fig. 26: Diagrama de flujo del servlet Login

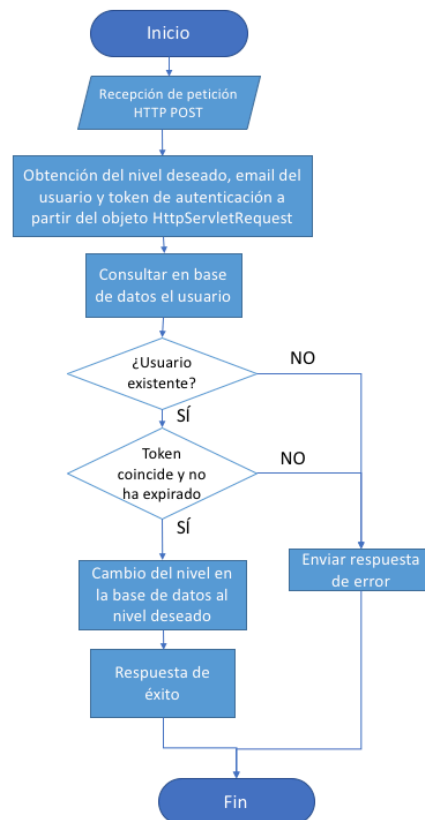


Fig. 27: Diagrama de flujo del servlet update

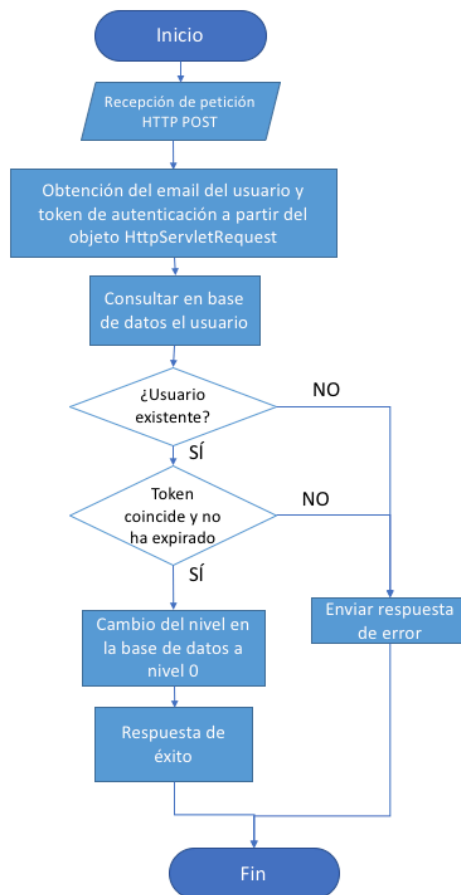


Fig. 28: Diagrama de flujo del servlet restart

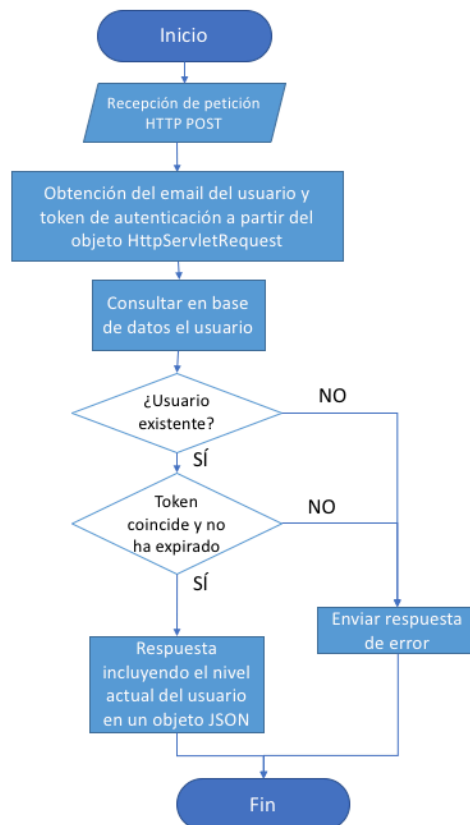


Fig. 29: Diagrama de flujo del servlet nivel

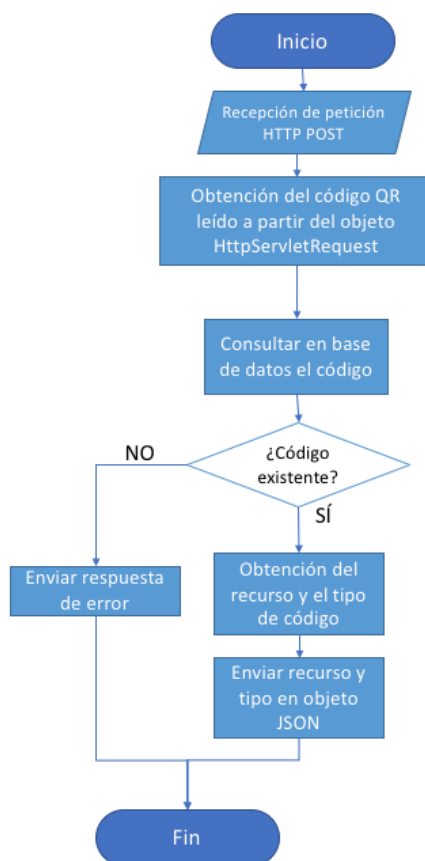


Fig. 30: Diagrama de flujo del servlet comprobarQR

#### 4.4. Implementación de la base de datos

La base de datos (EPSalumnos) diseñada en el apartado 3.3 cuenta con dos tablas independientes: Usuarios y QRcodes. Para la implementación de estas tablas es necesario definir el tipo de campo de cada variable así como la clave principal de cada tabla. La clave principal es un campo (o conjunto de campos) que identifican una fila de una tabla de la base de datos de forma unívoca. Por tanto, es irrepetible.

- Tabla **Usuarios**:
  - Campo *nombre*, del tipo VARCHAR, es decir, una cadena de longitud variable. Es necesario especificar una longitud máxima, se establece en 40 caracteres. Contiene el dato definido en el apartado 3.3 como “Nombre”
  - Campo *apellido*, del tipo VARCHAR. Longitud máxima 40 caracteres de nuevo. Contiene el dato definido en el apartado 3.3 como “Apellidos”
  - Campo *email*, del tipo VARCHAR y con longitud máxima de 40 caracteres. Será la clave principal de esta tabla: dos usuarios no pueden tener el mismo email, es un dato irrepetible. Contiene el dato definido en el apartado 3.3 como “Email”

- Campo *ultimoacceso*, del tipo DATETIME. Este tipo de datos representa una fecha y una hora concreta con el formato AA-MM-DD HH-MM-SS. Contiene el dato definido en el apartado 3.3 como “último acceso”
- Campo *nivel*, del tipo TINYINT. Este tipo de datos es un entero de un solo byte. Contiene el dato definido en el apartado 3.3 como “Nivel”
- Campo *token*, del tipo MEDIUMTEXT. Este tipo de datos ha sido el elegido debido a la gran longitud de los token de autenticación y así evitar problemas de truncamiento del token en caso de superar la longitud máxima. Contiene el dato definido en el apartado 3.3 como “Token”.
- Campo *tokenHora*, del tipo DATETIME ya explicado anteriormente. Contiene el dato definido en el apartado 3.3 como “TokenHora”.

En la Tabla 3 puede observarse la configuración de la tabla Usuarios de la base de datos.

| Columna             | Tipo de campo |
|---------------------|---------------|
| <b>nombre</b>       | VARCHAR(40)   |
| <b>apellido</b>     | VARCHAR(40)   |
| <b>email</b>        | VARCHAR(40)   |
| <b>ultimoacceso</b> | DATETIME      |
| <b>nivel</b>        | TINYINT       |
| <b>token</b>        | MEDIUMTEXT    |
| <b>tokenHora</b>    | DATETIME      |

Tabla 3: Campos de la tabla Usuarios

- Tabla **QRcodes**:

- Campo *id*, del tipo INTEGER UNSIGNED. A pesar de que no es un dato definido en el diseño de la base de datos, se incorpora al crear esta tabla. Nos servirá como clave principal de la tabla y será un valor que se autoincrementará en cada inserción.
- Campo *codigo*, del tipo VARCHAR con longitud máxima de 30 caracteres. Contiene el dato definido en el apartado 3.3 como “Código”
- Campo *recurso*, del tipo VARCHAR con longitud máxima de 100 caracteres debido a que puede contener enlaces de web largos. Contiene el dato definido en el apartado 3.3 como “Recurso”
- Campo *tipo*, del tipo TINYINT. Contiene el dato definido en el apartado 3.3 como “Tipo”

En la Tabla 4 puede observarse la configuración de la tabla QRcodes de la base de datos.

| Columna | Tipo de campo         |
|---------|-----------------------|
| id      | INTEGER UNSIGNED (40) |
| codigo  | VARCHAR(30)           |
| recurso | VARCHAR(100)          |
| tipo    | TINYINT               |

Tabla 4: Campos de la tabla QRcodes

Con esta definición en mente, la creación de estas tablas es intuitiva a través de los siguientes comandos en MySQL:

```
CREATE TABLE Usuarios (nombre VARCHAR(40), apellido VARCHAR(40), email VARCHAR(40) PRIMARY KEY, ultimoacceso DATETIME, nivel TINYINT, token MEDIUMTEXT, tokenHora DATETIME.);
```

```
CREATE TABLE QRcodes (id INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY, codigo VARCHAR(30), recurso VARCHAR(100), tipo TINYINT);
```

Una vez creadas las tablas, es necesario establecer quién se encargará de administrar cada una de ellas y quién, simplemente, accederá a ellas:

-La tabla Usuarios será una tabla de escritura y lectura para el servidor. Se encontrará inicialmente vacía y el servidor se encargará de ir rellenándola con los datos de los usuarios que inicien sesión.

-La tabla QRcodes será una tabla de lectura para el servidor. Los datos de esta tabla se encontrarán previamente insertados por el autor del presente trabajo. En ella se encontrarán todos los recursos asociados a los códigos QR creados.

El estado final de la tabla QRcodes se muestra en la Tabla 5.

| Código       | Recurso                 | Tipo | Explicación                                                                                                                    |
|--------------|-------------------------|------|--------------------------------------------------------------------------------------------------------------------------------|
| planta0info  | planta0.html            | 2    | Páginas web de información de las distintas plantas de la biblioteca. Tipo 2 debido a que son páginas web internas al servidor |
| planta1info  | planta1.html            |      |                                                                                                                                |
| planta2info  | planta2.html            |      |                                                                                                                                |
| planta3info  | planta3.html            |      |                                                                                                                                |
| plantaSinfo  | plantas.html            |      |                                                                                                                                |
| biblioteca01 | planta1biblioteca01.png | 0    | Imágenes del servicio de localización de las                                                                                   |
| biblioteca02 | planta1biblioteca02.png |      |                                                                                                                                |



|                   |                                     |   |                                                                                                                    |
|-------------------|-------------------------------------|---|--------------------------------------------------------------------------------------------------------------------|
| biblioteca03      | planta1biblioteca03.png             |   | plantas 0, 1 y 2.                                                                                                  |
| ...               | ...                                 |   | Tipo 0 por ser imágenes.                                                                                           |
| biblioteca59      | planta2biblioteca59.png             |   |                                                                                                                    |
| opina             | https://aplicaciones.uc3m.es...     | 1 | Páginas web de los distintos recursos accesibles mediante códigos QR. Tipo 1 por ser enlaces externos al servidor. |
| catalogo          | http://biblioteca.uc3m.es/...       |   |                                                                                                                    |
| normativaprestamo | https://www.uc3m.es/ss/Satellite... |   |                                                                                                                    |
| salasgrupo        | http://biblioteca3.uc3m.es/RES...   |   |                                                                                                                    |
| horario           | https://calendar.google.com/cal...  |   |                                                                                                                    |

Tabla 5: Estado final de la tabla QRcodes de la base de datos

#### 4.5. Implementación de conexión cliente-servidor

El cliente se comunica con el servidor mediante el uso de la librería *Volley* proporcionada por Android Studio. Esta librería proporciona varios métodos distintos para establecer la comunicación HTTP en función del tipo de respuesta esperada. A continuación se analiza cada uno de ellos y se especifica para qué se utilizaba cada uno en el presente trabajo:

- **JsonObjectRequest:** Esta petición obtiene como respuesta del servidor un objeto JSON en el cuerpo de la respuesta y permite añadir un objeto JSON a la petición por parte del cliente, por lo que es muy útil para intercambiar parámetros con el servidor en forma de objetos JSON tal y como se observa en la Fig. 31. Es utilizado para operaciones como el inicio de sesión, pedir la actualización del nivel de un determinado usuario o el reinicio del mismo.

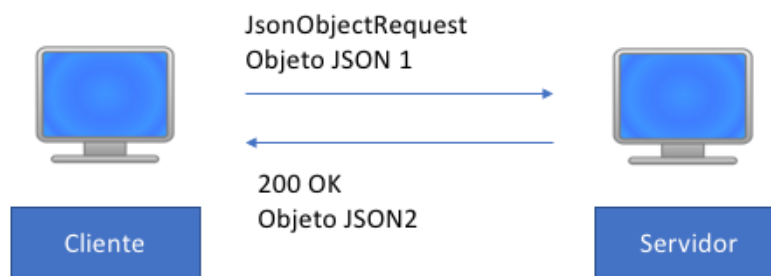


Fig. 31: Petición de objeto JSON con otro JSON adjunto

El contenido de los dos objetos JSON representados en la Fig. 31 se detalla a continuación para cada uno de los casos contemplados:

- **Inicio de sesión:** Tal y como ha sido diseñado el inicio de sesión, la aplicación tras autenticarse en los servicios de Google envía un token de autenticación al servidor. Dicho token solo se envía en caso de éxito de inicio de sesión. Sin

embargo, ha sido contemplado el envío de un token inválido ante posibles usos incorrectos. En las Tablas 6 y 7 se muestran los objetos JSON intercambiados para esta operación en caso de envío de un token válido y en el caso de uno no válido.

| JSON enviado |                | JSON recibido   |              |
|--------------|----------------|-----------------|--------------|
| Campo        | Valor          | Campo           | Valor        |
| Token        | (token válido) | codigoRespuesta | "Usuario OK" |

Tabla 6: Objetos JSON intercambiados para inicio de sesión válido

| JSON enviado |                  | JSON recibido   |             |
|--------------|------------------|-----------------|-------------|
| Campo        | Valor            | Campo           | Valor       |
| Token        | (token inválido) | codigoRespuesta | "No válido" |

Tabla 7: Objetos JSON intercambiados para inicio de sesión no válido

- **Comprobación de un código QR:** Al leer con el escáner un código QR se envía al servidor el contenido leído encapsulado en un objeto JSON. La respuesta por parte del servidor consiste en un objeto JSON con dos campos obtenidos de la base de datos: tipo y recurso. En caso de no haber coincidencia en la base de datos ambos campos serán devueltos con el valor -1. En las Tablas 8 y 9 se muestran los objetos JSON intercambios en caso de un código QR existente en la base de datos y otro no existente.

| JSON enviado |                                             | JSON recibido |                                                     |
|--------------|---------------------------------------------|---------------|-----------------------------------------------------|
| Campo        | Valor                                       | Campo         | Valor                                               |
| codigoQR     | (código QR existente, p.ej. "biblioteca01") | tipo          | ("0", "1" o "2")                                    |
|              |                                             | recurso       | (recurso asociado, p.ej. "planta1biblioteca01.png") |

Tabla 8: Objetos JSON intercambiados para comprobación de código QR existente

| JSON enviado |                                           | JSON recibido |       |
|--------------|-------------------------------------------|---------------|-------|
| Campo        | Valor                                     | Campo         | Valor |
| codigoQR     | (código QR no existente, p.ej. "abcdefg") | tipo          | "-1"  |
|              |                                           | recurso       | "-1"  |

Tabla 9: Objetos JSON intercambiados para comprobación de código QR no existente

- **Actualización de nivel:** Al superar el usuario un nivel del juego se envía un mensaje al servidor para que actualice el nivel guardado para dicho usuario en la base de datos. Con el fin de identificar al usuario y el nivel superado se adjunta un objeto JSON con el email del usuario y el nivel al que pasa el usuario. En las Tablas 10 y 11 se muestran los objetos JSON intercambiados en caso de que el usuario exista en la base de datos y en el caso contrario.

| JSON enviado |                                                   | JSON recibido |       |
|--------------|---------------------------------------------------|---------------|-------|
| Campo        | Valor                                             | Campo         | Valor |
| email        | (email existente, p.ej. "victorseoane@gmail.com") | respuesta     | "OK"  |
| nivelDeseado | (nivel a actualizar)                              |               |       |
| token        | (Token válido y no expirado)                      |               |       |

*Tabla 10: Objetos JSON intercambiados para actualizar el nivel de un usuario existente*

| JSON enviado |                                                 | JSON recibido |         |
|--------------|-------------------------------------------------|---------------|---------|
| Campo        | Valor                                           | Campo         | Valor   |
| email        | (email no existente, p.ej. "abacfds@gmail.com") | respuesta     | "ERROR" |
| nivelDeseado | (nivel a actualizar)                            |               |         |
| token        | (token válido y no expirado)                    |               |         |

*Tabla 11: Objetos JSON intercambiados para actualizar el nivel de un usuario no existente*

- **Reinicio de nivel:** Cuando un usuario elige la opción de "Nuevo juego", la aplicación manda al servidor una orden de reinicio de nivel. Esta petición lleva un objeto JSON con el email del usuario para poder identificar a dicho usuario. En las Tablas 12 y 13 se muestran los objetos JSON intercambiados en caso de que el usuario exista en la base de datos y en el caso contrario.

| JSON enviado |                                                   | JSON recibido |       |
|--------------|---------------------------------------------------|---------------|-------|
| Campo        | Valor                                             | Campo         | Valor |
| email        | (email existente, p.ej. "victorseoane@gmail.com") | respuesta     | "OK"  |
| token        | (token válido y no expirado)                      |               |       |

*Tabla 12: Objetos JSON intercambiados para reiniciar el nivel de un usuario existente*

| JSON enviado |                                                      | JSON recibido |         |
|--------------|------------------------------------------------------|---------------|---------|
| Campo        | Valor                                                | Campo         | Valor   |
| email        | (email no existente, p.ej. "victorseoane@gmail.com") | respuesta     | "ERROR" |

Tabla 13: Objetos JSON intercambiados para reiniciar el nivel de un usuario no existente

NOTA: En los anteriores ejemplos se ha considerado el envío de un token de autenticación válido y no expirado. En caso contrario, la respuesta del servidor sería "ERROR" en caso de token no válido o "EXPIRADO" en caso de token válido pero expirado.

- **ImageRequest:** Petición que obtiene una imagen en el cuerpo de la respuesta por parte del servidor (véase Fig.32). Al realizar la conversión de los datos de imagen a formato *Bitmap* es una opción muy cómoda y rápida de obtener imágenes del servidor sin necesidad de hacer conversiones ni otro tipo de operaciones, por lo que será utilizada para dichos fines.

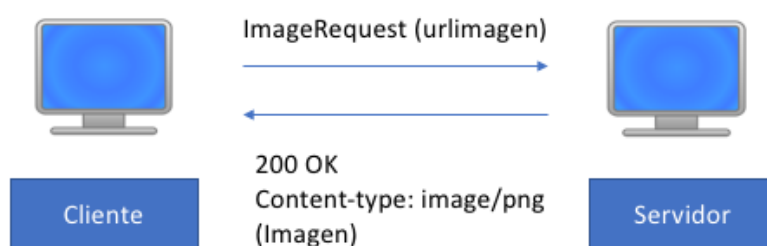


Fig. 32: Petición de imagen al servidor

#### 4.6. Implementación del escáner de códigos QR

El escáner de códigos QR es la herramienta básica del funcionamiento de la aplicación Android del cliente. A través de los códigos QR el usuario avanza en el juego, accede a diversos recursos y descubre información sobre la biblioteca. Por ello, el escáner de dicho códigos debe ser lo más rápido, estable e intuitivo posible.

La implementación manual de un escáner de códigos QR hubiera conllevado la aplicación de técnicas de reconocimiento de imágenes y *machine-learning*, además de una lectura a medida de todos los campos que presentan los códigos QR. Por este motivo y aprovechando la existencia de escáneres de código abierto, se optó por el uso del escáner incluido en la librería ZXing ("zebra crossing") [33]

Esta librería ofrece diversas opciones para la configuración del escáner como el tipo de códigos a escanear (de barras, QR u otras variantes), la orientación de la cámara y la reproducción de un sonido al reconocer un código. En esta implementación hemos

escogido códigos de tipo QR, orientación automática dependiendo de la orientación del teléfono móvil y no reproducir ningún sonido para ayudar al clima de silencio de la biblioteca. Teniendo en cuenta esta configuración y la documentación que ofrece la librería la forma de iniciar el escáner es la siguiente:

```
IntentIntegrator integrator = new IntentIntegrator(this);
integrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE);
integrator.setPrompt("Scan");
integrator.setOrientationLocked(false);
integrator.setCameraId(0);
integrator.setBeepEnabled(false);
integrator.setBarcodeImageEnabled(false);
integrator.initiateScan();
```

Estas instrucciones abren el escáner de códigos QR sobre una *activity* vacía que herede de la *activity* de captura definida en la librería. En este caso, dicha actividad en nuestro trabajo es *CaptureActivity*. Al escanear satisfactoriamente un código, la *activity* finaliza su ejecución y devuelve un resultado que puede ser recuperado mediante el método de Android *onActivityResult()*. El esquema de código en Java de recuperación y manejo de un código QR se muestra a continuación:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    IntentResult result =
    IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
    if(result!=null){
        if(result.getContents()==null){} //Hemos cerrado la camara
        else{ //Código leído
            String texto = (result.getContents()).toString();
            //Manejar el texto leído
        }
    }
}
```

El manejo del texto leído difiere según el contexto del escáner. Si el escáner ha sido abierto desde el menú principal la aplicación enviará dicho texto al servidor para comprobar el recurso que tiene asociado. En cambio, en algunas etapas del juego se usa un escáner QR en donde el resultado se maneja localmente y no es necesaria la intervención del servidor.

#### 4.7. Implementación de la conexión servidor-base de datos

*Java DataBase Connectivity* (JDBC) [34] es una interfaz de programación de aplicaciones (API) que permite operar sobre bases de datos desde aplicaciones Java. Está definido para cualquier sistema gestor base de datos y al ser destinado para Java, para cualquier sistema operativo.

Dependiendo del sistema gestor de base de datos empleado, se debe descargar y usar un driver JDBC distinto para lograr la conexión. Este driver no es más que una implementación específica de las interfaces para un sistema gestor de bases de datos dado. En el caso de MySQL, el gestor utilizado en este trabajo, recibe el nombre de *MySQL Connector/J* y puede descargarse en la página web de desarrolladores de MySQL (<https://dev.mysql.com>). Este archivo ha sido incluido en las librerías usadas para compilación y ejecución de los servlets que conectan con la base de datos.

También es necesario la creación de un *pool de conexiones*. Para ello se deben seguir los pasos de la documentación de Tomcat para este efecto, incluyendo la modificación del archivo de configuración *web.xml* y la creación de otro, *context.xml* dentro de los directorios WEB-INF y META-INF (respectivamente) en cada uno de los *servlets* que conecte con la base de datos.

Una vez hemos incluido el driver como librería para compilación y creado el *pool de conexiones*, podemos obtener un objeto del tipo *Connection*, especificando la dirección URL (*Uniform Resource Location*) de la base de datos. En nuestro caso, la base de datos se llama EPSalumnos y se encuentra en la misma máquina que el servidor y en el puerto 3306, puerto por defecto para MySQL. También es necesario especificar el usuario y contraseña de la base de datos.

```
try{
    String url = "jdbc:mysql://localhost:3306/EPSalumnos";
    Connection connection = DriverManager.getConnection(url, "root",
        "password");
}
catch (SQLException e){
    //Manejar excepción
}
```

A partir de este objeto *Connection* podemos obtener un objeto del tipo *Statement*. Sobre este objeto podemos realizar consultas SQL en la base de datos mediante los métodos *executeQuery()*, *execute()*, *executeUpdate()*. En el siguiente ejemplo se muestra la ejecución de una consulta SELECT:

```
try{
    Statement statement = connection.createStatement();
    String query = "SELECT FROM Usuarios WHERE nombre='Victor'";
    ResultSet rs = statement.executeQuery(query);
    connection.close();
}
catch (SQLException e){
    //Manejar excepción
}
```

Realizados estos pasos, es posible realizar cualquier tipo de consulta SQL sobre la base de datos alojada en el servidor. Las consultas ejecutadas en la base de datos desde el servidor son las siguientes:

-Comprobación de la existencia de un usuario en la tabla Usuarios: Esta consulta permite determinar si hay que hacer una inserción de un usuario o una actualización del ultimo acceso de dicho usuario a partir de un email dado.

```
SELECT * FROM Usuarios WHERE email= 'victorseoane96@gmail.com';
```

-Inserción de un nuevo usuario en la tabla Usuarios: Inserción de un nuevo usuario a partir de su nombre, apellidos y email, datos obtenidos del inicio de sesión. El nivel inicial es puesto a 1 y el último acceso toma el valor de la fecha y hora actuales mediante el uso de NOW().

```
INSERT INTO Usuarios (nombre, apellido, email, ultimoacceso, nivel, token, tokenHora) VALUES ('Victor', 'Seoane', 'victorseoane96@gmail.com', NOW(), 1, token, NOW());
```

-Modificación de un usuario para actualizar último acceso: Actualización de la fecha y hora del último acceso mediante el uso de NOW() de un usuario identificado por su email.

```
UPDATE Usuarios SET ultimoacceso=NOW(), tokenHora=NOW() WHERE email= 'victorseoane96@gmail.com';
```

-Reinicio de nivel: Reinicio de nivel de un usuario identificado por su email (nivel=1).

```
UPDATE Usuarios SET nivel=1, tokenHora=NOW() WHERE email= 'victorseoane96@gmail.com'
```

-Modificación de nivel: Actualización del nivel de un usuario identificado por su email a un nuevo nivel dado.

```
UPDATE Usuarios SET nivel=4, tokenHora=NOW() WHERE email= 'victorseoane96@gmail.com'
```

-Comprobación de un código QR: Obtención del recurso y el tipo de recurso asociado a un código QR dado.

```
SELECT recurso,tipo FROM QRcodes WHERE codigo='biblioteca01';
```



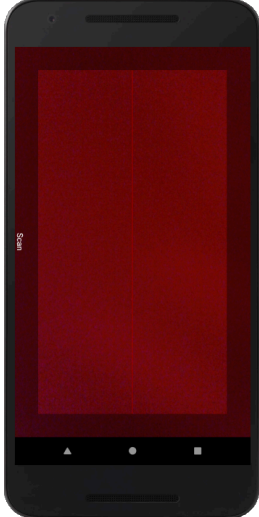

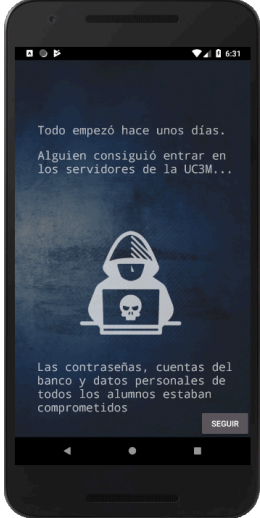
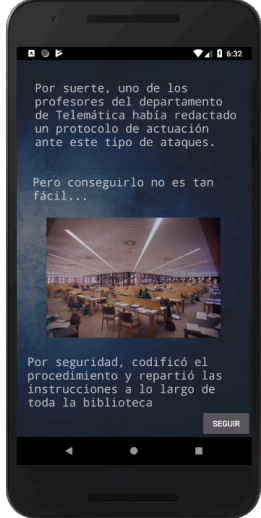



## 5. Resultados finales

Tras el proceso de diseño e implementación anteriormente descrito, se obtuvo la versión final de la aplicación, el servidor y la base de datos, así como de las conexiones entre ellos. Debido a que la parte principal del trabajo y la interfaz de comunicación con el usuario es la aplicación Android se muestra únicamente el resultado final de la misma tras el proceso completo de desarrollo. En la Tabla 14 se puede observar el nombre, descripción y capturas de pantalla de cada una de las *activities* que forman la aplicación.

| Nombre       | Descripción                                                                                    | Imagen                                                                               |
|--------------|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| SplashScreen | Pantalla de bienvenida a la aplicación.<br>Sirve de introducción y permite el inicio de sesión |   |
| MainActivity | Menú principal que muestra de forma visual todas las posibilidades que ofrece la aplicación    |  |

|                    |                                                                                                                                                                                         |                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| MenuBusquedaBiblio | <p>Permite buscar un código QR determinado a través de su etiqueta. De esta forma no es necesario escanearlo.</p>                                                                       |    |
| ServiciosScreen    | <p>Ofrece una lista de todos los servicios y recursos a los que se puede acceder desde la aplicación: catálogo, horario, reserva de salas, página web de la biblioteca entre otros.</p> |   |
| ImagenLocalizacion | <p>Muestra la localización de un código QR escaneado o buscado a partir de su etiqueta.</p>                                                                                             |  |

|                                 |                                                                                                           |                                                                                                                                                                            |
|---------------------------------|-----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CaptureActivity                 | Visualización a través de la cámara del escáner de códigos QR.                                            |                                                                                         |
| HelpActivity                    | Menú de ayuda para entender cada una de las opciones que se ofrece en el menú principal de la aplicación. |                                                                                        |
| Screen1,<br>Screen2,<br>Screen3 | Pantallas de introducción al juego de interacción en la biblioteca.                                       |   |

|  |  |                                                                                     |
|--|--|-------------------------------------------------------------------------------------|
|  |  |  |
|--|--|-------------------------------------------------------------------------------------|


*Tabla 14: Descripción y capturas de pantalla de las actividades de la aplicación*

Finalmente, en la Tabla 15 puede observarse la localización de una mesa en cada una de las plantas y en la Tabla 16 la página web desarrollada para explicar cada una de las plantas.

| Ubicación de la mesa | Imagen                                                                               |
|----------------------|--------------------------------------------------------------------------------------|
| Planta baja          |  |

|                |                                                                                     |
|----------------|-------------------------------------------------------------------------------------|
| Planta primera |   |
| Planta segunda |  |

Tabla 15: Capturas de pantalla de la localización de mesas de la biblioteca

| Planta        | Imagen                                                                               |
|---------------|--------------------------------------------------------------------------------------|
| Planta sótano |  |

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Planta baja</p>    |  <p>The smartphone screen displays the title "Estás en la Planta 0" (You are on the 0th floor). Below the title, it says "Esta es la planta de entrada a la biblioteca. Aquí encontrarás:" (This is the entrance floor of the library. Here you will find:). A list of facilities follows: "Baños" (Bathrooms), "Entrada" (Entrance), "Mostrador del centro de postgrado" (Postgraduate center counter), "Mostrador de información y referencia" (Information and reference counter), "Mostrador de préstamo" (Lending counter), "Zona de descanso" (Rest area), and "Zona de trabajo" (Work area). At the bottom, there is a floor plan diagram with various rooms labeled.</p> |
| <p>Planta primera</p> |  <p>The smartphone screen displays the title "Estás en la Planta 1" (You are on the 1st floor). Below the title, it says "Esta es la planta de lectura de la biblioteca. Aquí encontrarás además:" (This is the reading floor of the library. Here you will also find:). A list of facilities follows: "Manuales y otros archivos de OPAC" (Manuals and other OPAC files). Below the list, there is a floor plan diagram with various rooms labeled.</p>                                                                                                                                                                                                                        |
| <p>Planta segunda</p> |  <p>The smartphone screen displays the title "Estás en la Planta 2" (You are on the 2nd floor). Below the title, it says "Esta es la planta de estudio de la biblioteca. Aquí encontrarás:" (This is the study floor of the library. Here you will find:). A list of facilities follows: "Fondo especializado" (Specialized collection), "Revistas" (Magazines), "Oficina" (Office), "Salas individuales" (Individual study rooms), and "OPAC" (Online Public Access Catalog). At the bottom, there is a floor plan diagram with various rooms labeled.</p>                                                                                                                    |

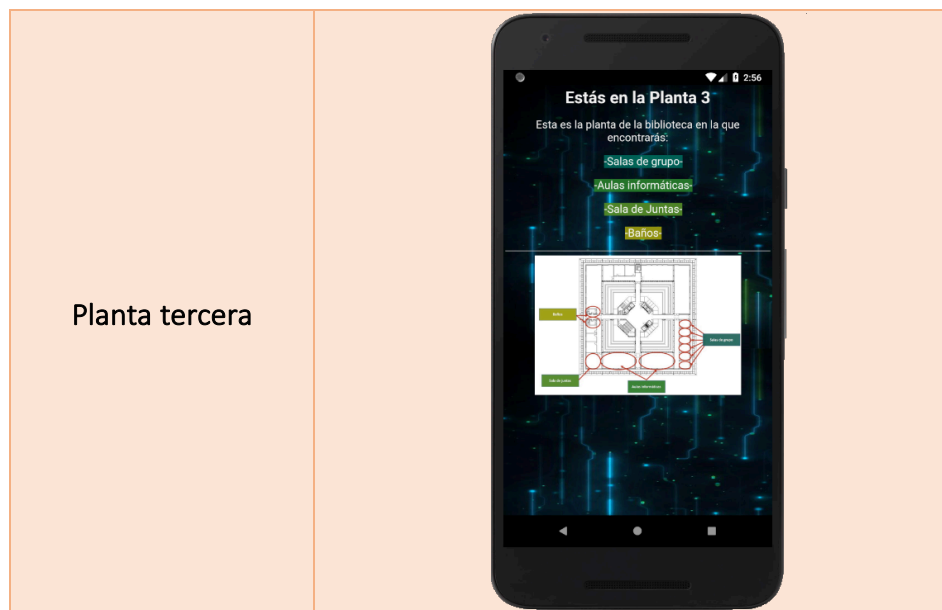


Tabla 16: Capturas de pantalla de las webs de información de las plantas de la biblioteca

## 6. Evaluación

Para evaluar el funcionamiento, el potencial y la facilidad de uso de la aplicación se llevó a cabo una prueba presencial el 12 de Junio de 2018 en la biblioteca de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

Diez alumnos voluntarios que habían instalado previamente la aplicación en sus teléfonos móviles participaron en esta prueba. Tras una pequeña introducción sobre las funcionalidades de la aplicación, se les invitó a que completaran el juego de pistas por la biblioteca y a usar el resto de funciones. El juego se desarrolló sin complicaciones técnicas ni logísticas y la experiencia fue descrita por los jugadores como entretenida y útil.

Al finalizar el juego, se les solicitó que rellenaran una hoja de evaluación de la experiencia que acababa de tener lugar. Dicha hoja puede observarse en el Anexo I de la presente memoria. La prueba tuvo una duración aproximada de 40 minutos y los resultados extraídos de la evaluación de los alumnos se muestran en la Tabla 17

|                                | Usuario 1 | Usuario 2 | Usuario 3 | Usuario 4 | Usuario 5 | Usuario 6 | Usuario 7 | Usuario 8 | Usuario 9 | Usuario 10 |
|--------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| Experiencia global de uso      | 9         | 10        | 10        | 9         | 9         | 8         | 10        | 7         | 8         | 6          |
| Facilidad de uso               | 7         | 9         | 9         | 8         | 9         | 9         | 9         | 9         | 7         | 9          |
| Utilidad juego de iniciación   | 9         | 6         | 6         | 9         | 9         | 9         | 9         | 7         | 9         | 6          |
| Utilidad localización de mesas | 8         | 9         | 10        | 8         | 8         | 8         | 10        | 8         | 8         | 8          |
| Utilidad acceso a servicios    | 9         | 10        | 9         | 8         | 10        | 8         | 8         | 8         | 6         | 8          |
| Utilidad aplicación            | 8         | 8         | 10        | 9         | 9         | 9         | 10        | 7         | 9         | 7          |
| Diseño estético                | 9         | 10        | 10        | 7         | 9         | 9         | 8         | 5         | 9         | 6          |
| Descargarías la aplicación     | Si        | Si        | Si        | Si        | Si        | Si        | Si        | No        | Si        | No         |

Tabla 17: Resultados de la prueba experimental

Como comentarios de mejora la mayoría de usuarios propuso la implementación de un juego de mayor duración con más niveles y preguntas más complicadas. Por otro lado, el aspecto más destacado por los usuarios fue el servicio de localización de las mesas y la originalidad de los distintos niveles del juego.

Como podemos observar en las Figuras 33 y 34, 8 de los 10 usuarios que participaron descargarían la aplicación para usarla diariamente y la experiencia global de uso de la aplicación fue muy positiva con una media de 8,6 sobre 10.

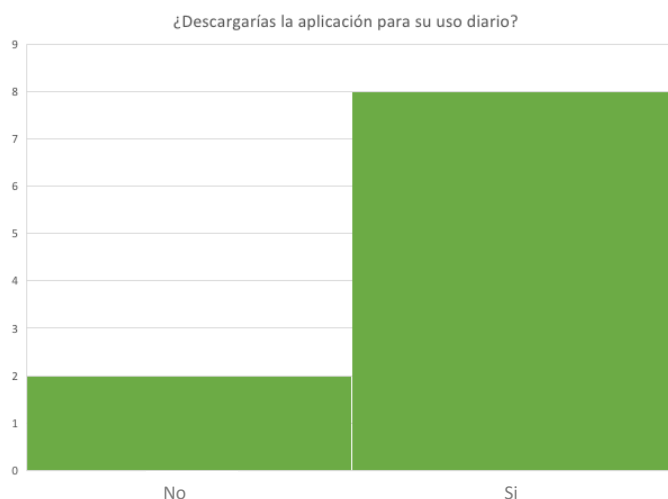
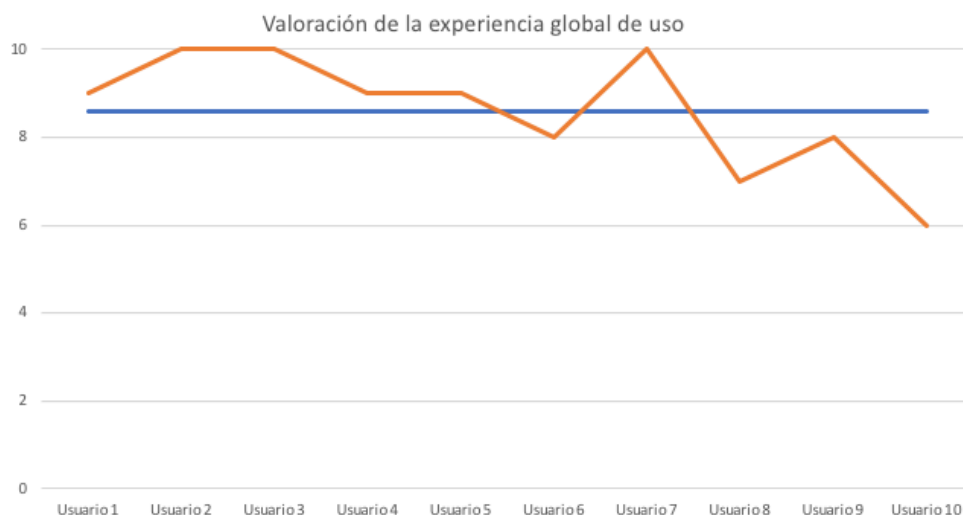


Fig. 33: Gráfico sobre la intención de uso diario de la aplicación





*Fig. 34: Gráfico sobre la valoración de la experiencia global de uso*

Cabe tener en cuenta que las personas que participaron en esta prueba ya eran alumnos de la universidad y que, por tanto, conocían la biblioteca. Por ello, ha de considerarse una mayor dificultad para alumnos de nuevo ingreso que necesiten ubicarse en la biblioteca antes de poder seguir las pistas del juego.

En las figuras 35, 36 y 37 pueden observarse algunas fotografías realizadas durante la prueba anteriormente descrita.



*Fig. 35: Fotografía nº. 1 de la prueba en la biblioteca*



*Fig. 36: Fotografía nº. 2 de la prueba en la biblioteca*



*Fig. 37: Fotografía nº. 3 de la prueba en la biblioteca*



## 7. Planificación y presupuesto

### 7.1. Planificación

En el presente capítulo se detalla la duración de cada una de las tareas que ha supuesto este proyecto.

Tal y como se desglosará a continuación la duración total de la realización del sistema propuesto (cliente, servidor, base de datos y conexiones) ha sido de aproximadamente 10 meses. Como puede observarse en la Tabla 18 dividimos este complejo proceso en tareas simples, especificando el tiempo que conllevó cada una de ellas.

|    | <b>Tarea</b>                                                              | <b>Inicio</b> | <b>Fin</b> |
|----|---------------------------------------------------------------------------|---------------|------------|
| 1  | Estudio y elección de tecnologías disponibles                             | 04/09/2017    | 11/09/2017 |
| 2  | Aprendizaje SQL                                                           | 15/09/2017    | 01/10/2017 |
| 3  | Aprendizaje programación Android                                          | 15/09/2017    | 19/10/2017 |
| 4  | Instalación de programas, máquina virtual, servidor...                    | 10/10/2017    | 15/10/2017 |
| 5  | Creación de la base de datos                                              | 03/11/2017    | 08/11/2017 |
| 6  | Diseño de las pantallas de la aplicación                                  | 10/11/2017    | 17/11/2017 |
| 7  | Diseño de la conexión cliente-servidor                                    | 20/11/2017    | 23/11/2017 |
| 8  | Diseño de la conexión servidor - base de datos                            | 25/11/2017    | 29/11/2017 |
| 9  | Diseño de inicio de sesión                                                | 01/12/2017    | 11/12/2017 |
| 10 | Diseño de las pantallas del juego                                         | 15/12/2017    | 25/01/2018 |
| 11 | Diseño de los servlets del servidor                                       | 03/01/2018    | 10/01/2018 |
| 12 | Diseño de códigos QR                                                      | 20/01/2018    | 27/01/2018 |
| 13 | Implementación conexión servidor – base de datos                          | 25/01/2018    | 04/02/2018 |
| 14 | Implementación de la pantalla de inicio de sesión                         | 01/02/2018    | 03/02/2018 |
| 15 | Implementación de servlet para inicio de sesión                           | 04/02/2018    | 07/02/2018 |
| 16 | Implementación conexión cliente – servidor                                | 10/02/2018    | 15/02/2018 |
| 17 | Implementación del menú principal                                         | 20/02/2018    | 24/02/2018 |
| 18 | Implementación del escáner de códigos QR                                  | 01/03/2018    | 10/03/2018 |
| 19 | Implementación de servlet para modificación del nivel en la base de datos | 15/03/2018    | 16/03/2018 |
| 20 | Implementación de la pantalla de ayuda                                    | 17/03/2018    | 18/03/2018 |
| 21 | Implementación de la pantalla de servicios                                | 20/03/2018    | 24/03/2018 |
| 22 | Implementación de la pantalla de búsqueda                                 | 30/03/2018    | 05/04/2018 |
| 23 | Implementación de las pantallas del juego                                 | 08/04/2018    | 12/05/2018 |
| 24 | Modificación de planos de la biblioteca                                   | 14/05/2018    | 15/05/2018 |
| 25 | Implementación del servicio de localización                               | 17/05/2018    | 20/05/2018 |
| 26 | Implementación de servlet para comprobar códigos QR                       | 18/05/2018    | 25/05/2018 |
| 27 | Corrección de errores                                                     | 01/06/2018    | 10/06/2018 |
| 28 | Prueba experimental                                                       | 11/06/2018    | 11/06/2018 |
| 29 | Redacción de la memoria                                                   | 25/04/2018    | 13/06/2018 |

*Tabla 18: Planificación de tareas*

A continuación se ofrece una pequeña descripción de cada una de las tareas para poder comprender de qué trata cada una de ellas.

**Tarea 1:** Búsqueda de tecnologías, programas y lenguajes de programación disponibles para la realización de los objetivos. Ya explorados, se lleva a cabo una elección justificada de las opciones usadas finalmente.

**Tarea 2:** Lectura de documentación y realización de prácticas con el lenguaje de consulta de bases de datos SQL. En concreto: estudio de los tipos de campos de MySQL, las distintas consultas disponibles y la correcta realización de las mismas.

**Tarea 3:** Lectura de documentación oficial y desarrollo de algunas aplicaciones simples de prueba, focalizando en el diseño de pantallas y las transiciones entre ellas, los distintos tipos de objetos y la estructura de una aplicación Android.

**Tarea 4:** Instalación en los equipos de los programas necesarios, configuración de la máquina virtual e instalación del servidor en ella.

**Tarea 5:** Estructuración de los datos a recoger y posterior creación de las distintas tablas que forman la base de datos con la correspondiente elección de los tipos de campos, las claves primarias y las relaciones entre las tablas.

**Tarea 6:** Elaboración de bocetos de las distintas pantallas que conforman la aplicación, en concreto ubicación de los botones, cuadros de texto y los distintos elementos que las forman.

**Tarea 7:** Elección de metodología utilizada para comunicar el cliente y servidor, así como la definición de los distintos parámetros que intercambian.

**Tarea 8:** Elección de la ubicación de la base de datos y definición de las operaciones de lectura/escritura por parte del servidor.

**Tarea 9:** Comprensión del modelo de autenticación de Google. Posterior diseño del flujo de datos con la correspondiente identificación de los parámetros intercambiados.

**Tarea 10:** Diseño de las distintas actividades y retos que forman el juego físico y diseño de las pantallas necesarias para el desarrollo del mismo.

**Tarea 11:** Identificación de funciones necesarias en el lado del servidor y división en distintos servlets independientes.

**Tarea 12:** Planificación de los distintos códigos QR necesarios para el funcionamiento de la aplicación y obtención de los mismos.

**Tarea 13:** Descarga del driver necesario para la conexión y programación de la conexión en Java. Realización de pruebas de su correcto funcionamiento.

**Tarea 14:** Implementación de la pantalla de inicio de sesión, visual y funcionalmente.

**Tarea 15:** Programación del servlet que se encarga del inicio de sesión: obtención de datos de Google, inserción en la base de datos y respuesta al usuario.

**Tarea 16:** Implementación de intercambios entre cliente y servidor.

**Tarea 17:** Implementación de la pantalla del menú principal, visual y funcionalmente

**Tarea 18:** Implementación de un escáner de códigos QR con librerías ya existentes. Posteriores pruebas de su correcto funcionamiento obteniendo la lectura en pantalla.

**Tarea 19:** Programación de los servlets que se encargan de actualizar y de reiniciar el nivel de un determinado usuario en la base de datos.

**Tarea 20:** Implementación de la pantalla de ayuda, visual y funcionalmente.

**Tarea 21:** Implementación de la pantalla de servicios, visual y funcionalmente.

**Tarea 22:** Implementación de la pantalla de búsqueda, visual y funcionalmente.

**Tarea 23:** Implementación visual y funcional de las distintas pantallas que conforman el juego. Integración con otros elementos como el escáner QR o los servicios.

**Tarea 24:** Modificación de los planos de la biblioteca para reflejar la realidad actual e incluir pequeñas anotaciones. Posteriormente, división de los mismos en zonas para la implementación del servicio de localización.

**Tarea 25:** Elaboración de la asociación entre códigos QR y zonas localizadas en los planes por medio de la tabla QRcodes de la base de datos.

**Tarea 26:** Programación del servlet que se encarga de comprobar la existencia de un código QR leído y devolver su recurso asociado.

**Tarea 27:** Testeo del correcto funcionamiento de la aplicación y corrección de los distintos errores producidos.

**Tarea 28:** Realización de una prueba experimental en el escenario de la biblioteca de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, tal y como se detalla en el siguiente capítulo.

**Tarea 29:** Redacción de la siguiente memoria recopilando el proceso al completo.

Por último, para finalizar este apartado, en la Fig. 38 se muestra un diagrama de Gantt con la ejecución de las tareas anteriormente descritas para ver de forma visual la organización del trabajo.

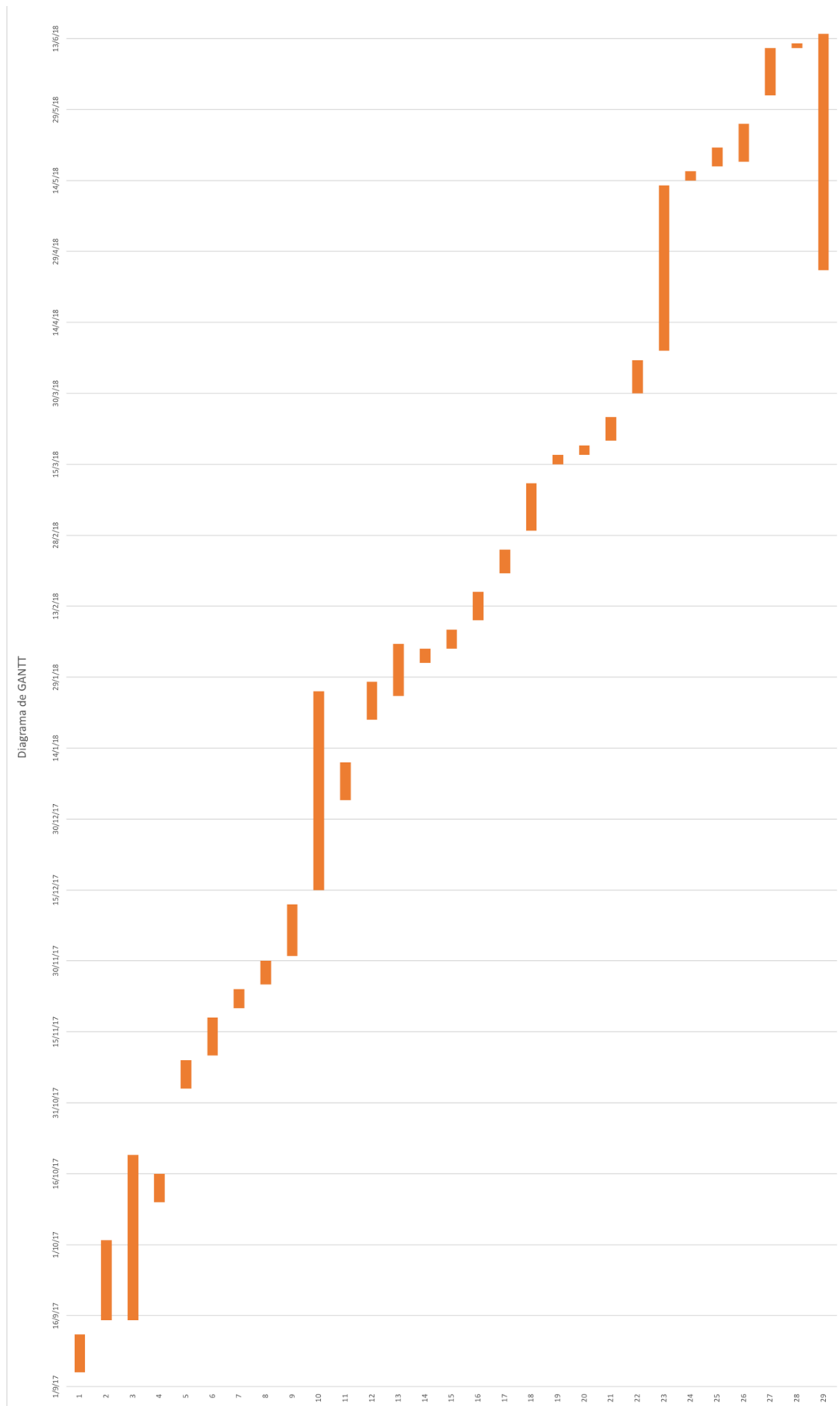


Fig. 38: Diagrama de Gantt



## 7.2. Presupuesto

Para elaborar un presupuesto asociado a la realización del proyecto propuesto, debemos analizar tanto la mano de obra como los medios tecnológicos y programas informáticos utilizados para su creación. A continuación, se detallan cada uno de ellos:

- **Software:** todos los programas informáticos utilizados para la realización del trabajo eran software gratuito a excepción de *Microsoft Office*, utilizado para la redacción de la presente memoria. Sin embargo, se disponía de una licencia educativa gratuita para dicho programa por ser parte de la institución Universidad Carlos III de Madrid, haciendo que tampoco compute en los costes del proyecto. Por ello, no hay ningún coste de software.
- **Hardware:** los dispositivos electrónicos utilizados han sido un ordenador de sobremesa, un ordenador portátil y un teléfono móvil Android. A fin de hacer cálculos de costes supondremos la vida útil de un ordenador igual a 6 años y la de un móvil Android a 3 años. Finalmente, considerando la duración total del proyecto como 10 meses, el precio medio de un ordenador como 900€ y de un móvil Android 300€ podemos establecer que el gasto total en hardware ha sido **333,33€**.

$$Total (HW) = 2 \cdot 900€ \cdot \frac{10 \text{ meses}}{6 \text{ años} \cdot 12 \text{ meses/año}} + 300€ \cdot \frac{10 \text{ meses}}{3 \text{ años} \cdot 12 \text{ meses/año}} = 333,33 \text{ €}$$

- **Mano de obra:** las horas dedicadas por cada miembro que ha participado en el trabajo han de ser remuneradas en función del puesto y título que ostentan. Así, utilizando las tablas salariales proporcionadas por la Universidad Carlos III de Madrid, podemos calcular el presupuesto de la mano de obra tal y como se observa en la Tabla 19.

| Persona física             | Categoría       | Coste por hora | Horas dedicadas | Total    |
|----------------------------|-----------------|----------------|-----------------|----------|
| Víctor Seoane Mérida       | Recién titulado | 23,43€         | 420             | 9840,60€ |
| Iria Manuela Estévez Ayrés | Titulado Doctor | 35,33€         | 30              | 1059,90€ |

Tabla 19: Presupuesto de mano de obra

Total de mano de obra= **10900,50€**

Por tanto, el presupuesto aproximado de realización del sistema propuesto al completo supondría un gasto total de **11233,83€**, repartido en 10900,50€ de mano de obra y 333,33€ de gasto en hardware.





## 8. Conclusiones y trabajo futuro

La biblioteca es un espacio esencial en la vida universitaria de la Universidad Carlos III de Madrid y su descubrimiento, exploración e integración puede suponer un reto para los nuevos alumnos, que pueden llegar a sentirse desubicados.

Este trabajo busca hacer más liviana esta tarea, simplificando los espacios y convirtiéndolos en el escenario de un juego que, más allá de un juego, resulta ser un aprendizaje. Un aprendizaje de todos los espacios, posibilidades y herramientas que ofrece la biblioteca. Una vez finalizado el juego la vida de la aplicación no termina, permitiendo el acceso a recursos y herramientas de la biblioteca y ofreciendo el servicio de localización dentro de ella.

Si bien es cierto que está planificado un tour por la biblioteca durante la asignatura de Técnicas de búsqueda y uso de la información, en ocasiones no es hasta el mes de octubre o incluso noviembre cuando tiene lugar y los alumnos se familiarizan con el espacio de la biblioteca. Sin embargo, este trabajo no busca ser un sustituto de dicha actividad, sino un complemento, un apoyo que haga de la actividad algo más ameno, memorable y dinámico o que simplemente se desarrolle de forma independiente al desarrollo de la asignatura y pueda realizarse con anterioridad.

Como podemos ver en la evaluación, el sistema ha tenido una buena aceptación por parte de los alumnos que se sometieron a la prueba y una gran mayoría piensa que podría ser útil para los nuevos alumnos, alcanzando los objetivos buscados.

Esta primera versión tiene limitaciones que podrían ser solventadas en el futuro. La más importante de ellas y que ya ha sido comentada anteriormente es la necesidad de uso de HTTPS ya que solo con la implantación de este protocolo podemos conseguir seguridad para el intercambio de información entre el cliente y el servidor. La decisión de no implantar este protocolo va muy ligada a la siguiente limitación: no se ha distribuido públicamente la aplicación. En la actualidad, el fichero de instalación de la aplicación (fichero .apk) ha sido enviado a los evaluadores por medio de correo electrónico sin dependencia alguna de la tienda de aplicaciones de Google. Es por ello que solo los receptores de dicho correo pueden instalar la aplicación y resulta razonable justificar la ausencia de HTTPS debido a que se trata de un sistema cerrado.

Por otra parte, el sistema de localización dentro de la biblioteca podría mejorarse prescindiendo de los códigos QR y utilizando GPS (*Global Positioning System*). Esto supondría la calibración del sistema de coordenadas dentro de la biblioteca y la definición de un plano en torno a las mismas.

Otras posibles mejoras, teniendo en cuenta la opinión de los usuarios, abarcan desde una sofisticación de la interfaz gráfica de la aplicación (añadiendo movimiento, dinamismo, vídeos y otros elementos gráficos) hasta una ampliación de los niveles del juego (nuevos enigmas, pasatiempos y preguntas que resolver). Ligado con esto último y

pensando en un uso mayor de la aplicación, podría llegar a ser necesario establecer varios recorridos alternativos para el juego, de forma que los alumnos se encontraran distribuidos entre los distintos espacios y no se crearan cuellos de botella.

Finalmente, para alcanzar al mayor número de usuarios posibles tal y como se pretende, se podría desarrollar una versión de la aplicación para sistemas móviles cuyo sistema operativo es iOS, consiguiendo así un alcance casi completo de alumnos.

Este trabajo podría incluso servir como base para la exploración de otros lugares de la universidad mediante el uso de tecnología de códigos QR o para implantar un servicio de localización en la totalidad del campus, favoreciendo y facilitando el encuentro entre alumnos.

## Bibliografía

[1] ITGP Privacy Team. *EU General Data Protection Regulation (GDPR): An Implementation and Compliance Guide*, 2ª edición. Europa: IT Governance Publishing, 2017. [En línea]. Disponible en: <https://www.safaribooksonline.com/library/view/eu-general-data/9781849288378/> [Accedido: 05-03-2018]

[2] C. García Murillo, “Aplicación práctica (y progresiva) del nuevo Reglamento europeo de protección de datos”, *Lawandtrends*, 2016. [En línea]. Disponible en: <http://www.lawandtrends.com/noticias/tic/aplicacion-practica-y-progresiva-del-nuevo.html> [Accedido: 09-03-2018]

[3] M. Pérez et al., “Discovering the campus together: A mobile and computer-based learning experience”, *Journal of Network and Computer Applications*, vol. 35, n.º 1, pp. 176-188, 2011. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S108480451100049X>. [Accedido: 15-04-2018].

[4] M. Kelly Schultz, “A case study on the appropriateness of using quick response (QR) codes in libraries and museums”, *Library & Information Science Research*, vol. 35, n.º 3, pp. 207-215, 2013. [En línea]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0740818813000315>. [Accedido: 15-04-2018].

[5] R. Ashford, “QR codes and academic libraries: Reaching mobile users”, *College and Research Libraries News*, vol. 71, n.º 10, pp. 526-530, 2010. [En línea]. Disponible en: <https://crln.acrl.org/index.php/crlnews/article/view/8454>. [Accedido: 22-04-2018].

[6] “The client/model server”, *IBM Knowledge Center*. [En línea]. Disponible en: [https://www.ibm.com/support/knowledgecenter/en/SSAL2T\\_8.1.0/com.ibm.cics.tx.doc/concepts/c\\_clnt\\_sevr\\_model.html](https://www.ibm.com/support/knowledgecenter/en/SSAL2T_8.1.0/com.ibm.cics.tx.doc/concepts/c_clnt_sevr_model.html) [Accedido: 03-04-2018]

[7] G. Fernández Pérez, *iOS: todo lo que siempre has querido saber sobre tu iPhone y iPad*. España: Gerardo Fernández Pérez, 2013. [En línea]. Disponible en: <https://itunes.apple.com/es/book/ios-todo-lo-que-siempre-has-querido-saber-sobre-tu/id768222195?mt=11> [Accedido: 19-03-2018]

[8] H. McCracken, “Who’s winning, iOS or Android? All the numbers, all in one place”, *Time*, abr. 2013. [En línea]. Disponible en: <http://techland.time.com/2013/04/16/ios-vs-android/> [Accedido: 05-04-2018].

[9] “The Swift Programming Language”, *Apple developer*. [En línea]. Disponible en: [https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/](https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/) [Accedido: 05-04-2018].

- [10] J. Tomas Gironés. *El gran libro de Android*, 3ª edición. España: Marcombo S.A, 2013.
- [11] A. Ramos y M Ramos, *Aplicaciones web* , 2ª edición. España: Paraninfo, 2014.
- [12] J. Cuello y J. Vittone. “Lanzando la app”, en *Diseñando apps para móviles*. España: 2013, José Vittone, páginas 231-244. [En línea]. Disponible en: <http://appdesignbook.com/es/contenidos/publicar-app-en-tienda/> [Accedido: 05-04-2018].
- [13] “An update on Eclipse Android Developers Tools”, Android Developers Blog [En línea]. Disponible en: <https://android-developers.googleblog.com/2015/06/an-update-on-eclipse-android-developer.html> [Accedido: 06-04-2018].
- [14] “Android Studio: User Guide”, *Android Developers* [En línea]. Disponible en: <https://developer.android.com/studio/intro/> [Accedido: 06-04-2018].
- [15] “¿Qué es Gradle en Android Studio?”, *AndroidStudioFAQs*. [En línea]. Disponible en: <https://androidstudiofaqs.com/conceptos/que-es-gradle-en-android-studio> [Accedido: 06-04-2018].
- [16] “NetBeans IDE – Overview”, *NetBeans*. [En línea]. Disponible en: <https://netbeans.org/features/index.html> [Accedido: 06-04-2018]
- [17] “NBAndroid – plugin detail”, *NetBeans*. [En línea]. Disponible en: <http://plugins.netbeans.org/plugin/19545/nbandroid> [Accedido: 06-04-2018]
- [18] A. Kumar Bansal, *Introduction to Programming Languages*, 1ª ed. EEUU: CRC Press, 2014.
- [19] “Tiobe Index”, *Tiobe*. [En línea]. Disponible en: <https://www.tiobe.com/tiobe-index/> [Accedido: 20-04-2018].
- [20] “¿Qué es Django?”, *Django Girls Tutorials*. [En línea]. Disponible en: <https://tutorial.djangogirls.org/es/django/>. [Accedido: 20-04-2018].
- [21] J. Goodwill y A. Vukotic, “Introduction to Apache Tomcat 7” en *Apache Tomcat 7*, 1ª edición. Apress, 2011. [En línea]. Disponible en: <https://www.safaribooksonline.com/library/view/apache-tomcat-7/9781430237235/> [Accedido: 30-03-2018]
- [22] “¿Qué es una base de datos relacional?”, *Amazon Web Services* [En línea]. Disponible en: <https://aws.amazon.com/es/relational-database/> [Accedido: 07-04-2018]
- [23] S. Sumathi y S. Esakkirajan, *Fundamentals of Relational Database Management Systems*, 2ª edición. Polonia: Springer, 2010.
- [24] C. Fehily, *SQL: Visual Quickstart Guide* ,3ª edición. EEUU: Peachpit Press, 2008.

- [25] L. Ullman, *MySQL: Visual Quickstart Guide*, 2ª edición. EEUU: Peachpit Press, 2006. Disponible en: <https://www.safaribooksonline.com/library/view/mysql-second-edition/0321375734/> [Accedido: 10-04-2018]
- [26] C. Newman, *SQLite*. 1ª ed. EEUU: Sams Publishing, 2004. [En línea]. Disponible en: <https://www.safaribooksonline.com/library/view/sqlite/067232685X/>
- [27] “About PostgreSQL”, *PostgreSQL.org*. [En línea]. Disponible en: <https://www.postgresql.org/about/> [Accedido: 07-04-2018]
- [28] “Transmitting Network Data using Volley”, *Android Developers*. [En línea]. Disponible en: <https://developer.android.com/training/volley/index.html>. [Accedido: 22-04-2018].
- [29] B. Smith, *Beginning JSON*, 1ª ed. EEUU: Apress, 2015. [En línea]. Disponible en: <https://www.safaribooksonline.com/library/view/beginning-json/9781484202029/> [Accedido: 10-04-2018]
- [30] “Authenticate with a backend server”, *Google Developers*. [En línea]. Disponible en: <https://developers.google.com/identity/sign-in/android/backend-auth>. [Accedido: 03-02-2018]
- [31] “Inicio de sesión con Facebook para Android: inicio rápido”, *Facebook for developers*. [En línea]. Disponible en: <https://developers.facebook.com/docs/facebook-login/android>. [Accedido: 03-02-2018]
- [32] “Infografía: ¿cómo funciona el código QR?”, *Merca 2.0*. [En línea]. Disponible en: <https://www.merca20.com/infografia-como-funciona-el-codigo-qr/> [Accedido: 20-04-2018]
- [33] “Zxing”, *Github*. [En línea]. Disponible en: <https://github.com/zxing/zxing> [Accedido: 05-05-2018]
- [34] G. Reese, “Introduction to JDBC” en *Database Programming with JDBC & Java*, 2ª edición. O’Reilly media Inc., 2000, 25-55- [En línea]. Disponible en: <https://www.safaribooksonline.com/library/view/database-programming-with/1565926161/> [Accedido: 09-04-2018]

## Summary

### Introduction

On the first days at the university it may become difficult to orientate yourself inside the library, to be aware of the services it offers or to understand what is each space of the library meant to. Such a diverse and mixed space with so many needs to be satisfied needs a powerful tool to inform its users about its resources, its possibilities and its infrastructure. That is why this bachelor thesis is focused on the creation of a tool to make it easier, funnier and more attractive. The chosen tool for this purpose is an application for the Android platform.

This application consist of a challenge divided into levels where the user has to use information about the library in order to resolve different enigmas and to answer some questions. In addition, looking for the total interaction between the user and the physical space of the library, QR codes has been used and they must be scanned by the players at different moments during the game.

The main objective of this work is the development of a game for the Android platform which allows the user to discover and explore every service offered by the library. Each user's game progress is different so a method to authenticate the users is needed. Moreover, for saving the progress of every user of the game and for offering the user the option to pause the game and resume it later, a web server and a database must be implemented.

We can summarize all the objectives of this bachelor thesis as follows:

- Choice of a programming language for the developing of the Android application.
- Design and implementation of a game for the Android platform based on the library of the University Carlos III of Madrid.
- Implementation of an authentication system and an access control.
- Development of a database to store the users' data, the level of the game in which they are, the QR codes and the images.
- Choice of a web server technology which fulfils the requirements.
- Implementation of a web server to authenticate the users, to provide images and webpages and to store users' information.

The system described above handles users' data so the processing of the data must be analysed to fulfil the requirements of the GDPR (General Data Protection Regulation) settled by the European Union. According to this regulation, the security measures have to grant a level of security commensurate with the risk. The data used by the application are basic so the risk level is low and the required security measures are granted by storing the data on a secure server of the University Carlos III of Madrid.

## State of art

Android applications are undergoing a really strong rise which has produced the emergence of new technologies, development environments and programming options. The usage of QR codes for offering information is a common fact and nowadays more and more institutions use them for that purpose. The following cases are interesting for the developing of this thesis' app:

- **University Pompeu Fabra:** In the year 2010 in the University Pompeu Fabra of Barcelona (Spain), an experiment with new students was carried out with the goal of exploring the campus using three different ways: a web page, a NFC reader installed on a smartphone and visual exploration. Students had to explore the campus, get information about the areas and answer a questionnaire.

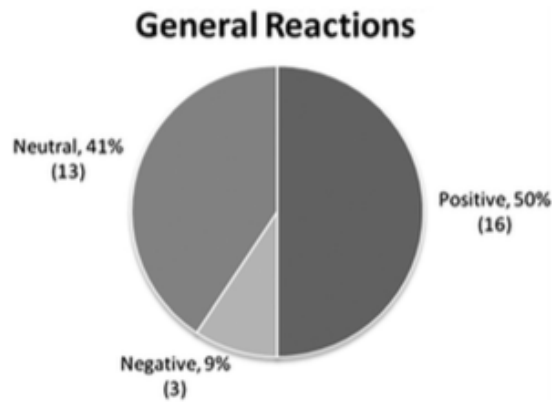
46 NFC codes were allocated along the campus, containing images, audios and videos explaining the different areas of the campus. Those contents were also available on the web page and posters. The investigation showed that the NFC technology was easily adopted by the students. Besides, the students which used the NFC reader instead of the other ways got a better mark in the questionnaire and were more original in their answers.

Students described the experience like funny, innovative and useful. The university showed determination to continue using new technologies such as NFC or GPS for goals like the one described.

- **Ryerson University and Museum of Inuit Art:** In the year 2013 in Toronto (Canada), two systems based on QR codes where tested.
  - The Ryerson University used QRcodes for different purposes such as codes to access to an audio guide, to open the web pages of the resources of the library or to examine the catalogue of books.
  - The Museum of Inuit Art placed QR codes next to every piece of art. Once the code was scanned, some information was showed, like a map with the origin country of the artist, a video of the author creating the piece or a style analysis. Moreover, those who scanned the codes were able to leave comments of the piece of art and there were some codes to follow the social networks of the museum.

56 people of the two institutions where interviewed and the results showed that the general reaction to the QR codes was mainly positive as it is shown in Fig. 39.





*Fig. 39: General reaction of the interview to the QR codes (Figure taken from [4])*

More people confessed having used QR codes in the University (56%) than in the Museum (25%). Finally, the students of the library were less aware of the existence of the QR codes than the visitors of museum. This fact shows that the promotion of the system is a very important part of the project.

The system of this thesis follows the model Client-Server, where the tasks are distributed between the services provider (**server**) and the requesters of those services (**client**). It is important to consider the existing technologies for the developing of the client, the server, the database and the connections between them:

For the **client's development**, a platform for the application and a development environment needed to be chosen.

- Platform: iOS, Android and the web platform were taken into account. Web platform option was dismissed because of the difficulty of implementing a game using HTML. Between Android and iOS the chosen one was Android because it has a larger scope of users and an easier distribution via Google Play.
- Development environment: the choice was between Android Studio and NetBeans. Both of them are powerful tools and they offer similar functionalities. However, the documentation of Android Studio is much wider and it offers an easier integration with other products of Google (such as Google Sign-in) so it was the final choice.

For the **server's development**, it was necessary to choose a programming language and a server technology.

- As it is shown in Fig. 40, the most popular programming languages are Java, C, C++ y Python. Since C and C++ are not often used for server programming, the final options were Python and Java. Finally, the chosen programming language was Java because an earlier knowledge by the author and a greater simplicity of the tools associated with the language, such as the servlets and Apache Tomcat.

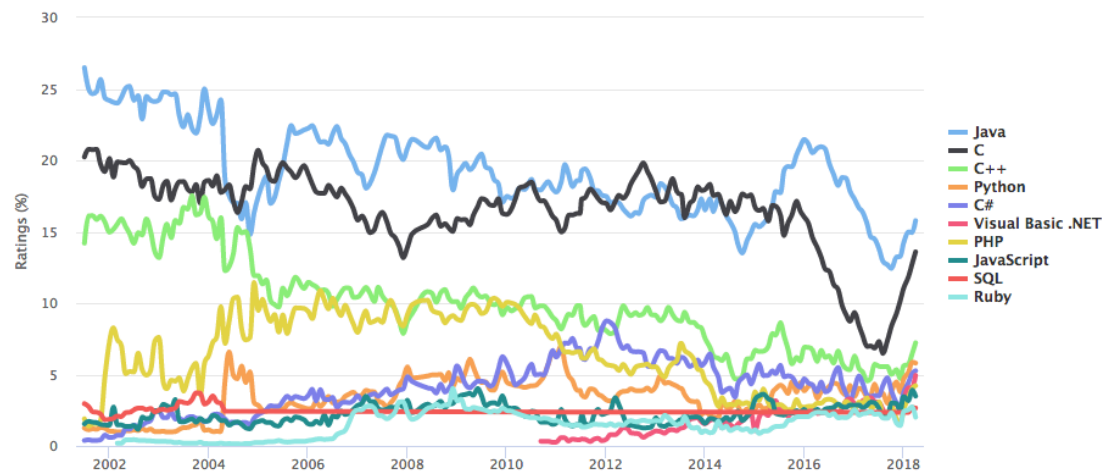


Fig. 40: Popularity of the programming languages (Figure taken from [19])

- Apache Tomcat was the chosen tool for the developing of the server. It is a Java Servlet Container and it works receiving the requests and rerouting them to the servlets. No other tool was considered because its great simplicity and potential.

For the **database's deployment** a Relational DataBase Management System (RDBMS) had to be chosen. MySQL, a free RDBMS was the final choice because its simplicity, its optimization for reading queries and its ease of use. Other contemplated alternatives were SQLite and PostgreSQL and, although they were also viable for the database of the system, they were more sophisticated than needed.

For the **connection between client and server**, Volley was selected as the HTTP (HyperText Transport Protocol) library used for communicating client and server. It is a simple and quick open-source tool that allows a simple connection between an Android application and a HTTP server. The data exchanged follows the JSON (JavaScript Object Notation) format, composed of name-value pairs or lists of values. It is a interoperable format and it is really straightforward to work with it in Java. Finally, for the user login, Facebook Login, Google Sign-In and own methods of authentication were studied and Google Sign-In appeared to be the best option, making the process simpler and having the largest scope because every student of the university has a Google account.

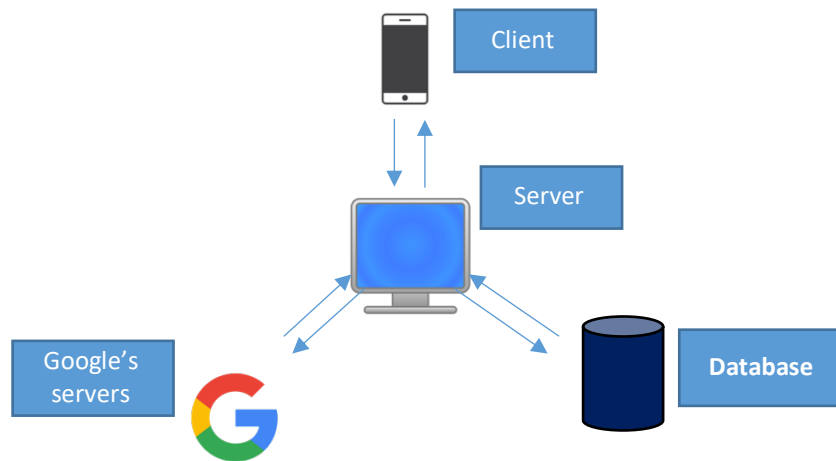
A **QR code** is a two-dimensional barcode which stores codified data (text, webpages, locations, images...). It has become a great instrument for showing information to the clients and they are nowadays the main tool of interaction between the smartphone and the environment. In this thesis they are used for offering information about the floors of the library, accessing some resources of the library and simulating a tables' location system.

## Design

### Server's requirements:

- Communication with the client for data exchange.
- Extraction of login data using Google's services.
- Storage and update of login data.
- Image providing.
- Database update for changing the level of a user.
- Verification of the resources linked to a given QR code and subsequent return to the client

So, the role of the server is to answer the requests made by the client, checking the database's tables and keeping them updated. This server has to validate login attempts using Google services, store the login information, modify the users' level and return the resources linked to a QR code. The design of the server is shown in Fig. 41.



*Fig. 41: Server's design*

### Client's requirements:

- Login screen with a pop-up for wrong login attempts.
- Authentication using Google's services.
- Sending of the authentication token to the server.
- Requests and updates of the current level.
- Main menu screen with a button for logging out.
- Help screen explaining all the functionalities.
- Screen with direct links to some resources of the library.
- Images' requests to the server.
- Resuming the game at the current level.
- Scanning QR codes and sending them to the server.

The most important thing when designing the client is to make a list of the screens it needs and to create a mock-up of them:

- **Login screen**, composed of the logo of the University Carlos III of Madrid and a button to sign-in using Google. In case of wrong login it shows a pop-up informing about the situation.
- **Main menu screen** shows the six categories of the application using circular buttons with icons representing the categories. These six categories are: new game, resume, scan, services, search and help.
- **QR reader screen** opens the device's camera and looks for QR codes. This screen was not designed because it was provided by the library used for the QR reader (ZXing library)
- **Table's location screen** shows a plan of the library with the table where the QR code was scanned marked in red.
- **Services screen** shows a list of the services of the library which can be directly accessed by clicking the buttons.
- **Search screen** makes possible to look for a table's location using its tag instead of scanning its QR code.
- **Help screen** explain each one of the six categories of the main menu screen.

Database's requirements:

- A table for storing users' information.
- A table for storing QR codes, their linked resources and their type.

The data to be stored and the database's design can be seen in the Fig. 42.

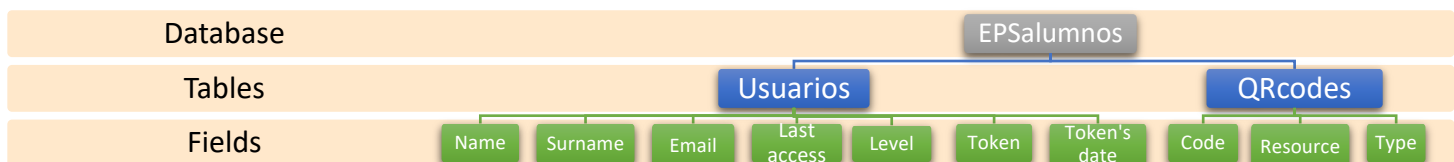


Fig. 42: Database's design

The connection between client and server starts with the login action. Its process is the following:

1. The application sends the user and the password to Google's services.
2. Google sends a success answer including an authentication token valid for 30 minutes.
3. The application sends this token to the server.
4. The server gets the user's info from Google using the token.
5. The server stores the user's info in the database.
6. Server informs the client about the success of the operation.

After this has happened, the answers of the server will depend on the client's requests. For identifying the user that makes the request, the email and the token is attached to every request. For security, the validity of the token is only 30 minutes but

this time is refreshed when receiving a new valid request. Finally, it is necessary to say that for achieving the total security of the system HTTPS (HyperText Transport Protocol Secure) should be used.

## Implementation

The devices and programs used for this project are listed below:

### Devices:

- Desktop computer: iMac, late 2013
  - Intel Core i5 processor, 2.7 GHz
  - 8GB 1600 MHz DDR3 memory
  - Intel Iris Pro 1536 MB graphics
  - 1 TB hard disk
  - Operative system: macOS High Sierra
- Laptop computer MacBook Air, early 2015.
  - Intel Core i5 processor, 1.6 GHz
  - 4GB 1600 MHz DDR3 memory
  - Intel HD Graphics 6000 1536 MB graphics
  - 128 GB hard disk
  - Operative system: macOS High Sierra

### Programs::

- Android Studio 3.1 for macOS: Development environment used for the creation of the Android application.
- VirtualBox 5.0.4: Virtual box which emulates the Linux operative system. The requirements of this virtual box were:
  - Operative system: Ubuntu 16.04 (64 bits)
  - Base memory: 2232 MB
  - Video memory: 22 MB
  - Virtual size: 9 GB

In detail, the programs used inside this Linux virtual box were:

- Eclipse 3.8: programming and compilation in Java of the servlets of the server.
- Apache Tomcat 8.0.28: Servlet container used for the server's launch.
- MySQL 5.7.22: management of the server's database.
- Microsoft Word 2016 for macOS: text editor for the writing of the report.

The application was structured into the following activities:

- MainActivity: it shows the main menu screen and manage the results of the QR reader, the requests of level of the user and the request of reset of level.

- SplashScreen: it shows the login screen and manage the connection with the server to authenticate the user.
- ServiciosScreen: it shows a list of the resources of the library and eases the access to them.
- ImagenLocalizacion: it shows the location image of a table of the library and allows the user to share it.
- HelpActivity: it explains each functionality of the main menu.
- MenuBusqueda: it offers a list of QR codes used to locate tables, so it is possible to see the location of a table without scanning its QR code.
- CaptureActivity: it allows the opening of the QR reader.
- Screen1, Screen2, Screen3: they show the introduction to the game.
- Other activities for the implementation of the levels of the game.

To fulfil the requirements of the server, it was structured as follows:

- **Path webapps/comandos:** it stores the servlets developed to fulfil the design's requirements. The compiled Java files (.class) are found in the path **WEB-INF/classes**, while the necessary libraries are in **WEB-INF/lib**.

Servlets are reachable through the following URL: <http://IPAddressOfTheServer:8080/comandos/Servlenamet>. There are 5 servlets:

- Servlet **login**: it checks the user's credentials and updates the database.
  - Servlet **update**: it updates the level of a user in the database. Previous to that, it verifies the authentication token that is attached to the request.
  - Servlet **restart**: it works as the previous one but updating the level of the user to 0.
  - Servlet **nivel**: it returns the level of a given user.
  - Servlet **comprobarQR**: it checks if the received QR codes is present in the database. If so, returns the information linked to that code.
- **Path webapps/media:** it stores the images which must be allocated in the server. They are reachable through the following URL: <http://IPAddressOfTheServer:8080/media/imageName>.
- **Path webapps/web:** it stores the HTML files which must be offered directly to the application. They are reachable through the following URL: <http://IPAddressOfTheServer:8080/web/fileName>.

The database's final tables are shown in the Tables 20 and 21.

| Field               | FieldType   |
|---------------------|-------------|
| <b>nombre</b>       | VARCHAR(40) |
| <b>apellido</b>     | VARCHAR(40) |
| <b>email</b>        | VARCHAR(40) |
| <b>ultimoacceso</b> | DATETIME    |
| <b>nivel</b>        | TINYINT     |
| <b>token</b>        | MEDIUMTEXT  |
| <b>tokenHora</b>    | DATETIME    |

*Tabla 20: Fields of the table Usuarios*

| Field          | FieldType             |
|----------------|-----------------------|
| <b>id</b>      | INTEGER UNSIGNED (40) |
| <b>codigo</b>  | VARCHAR(30)           |
| <b>recurso</b> | VARCHAR(100)          |
| <b>tipo</b>    | TINYINT               |

*Tabla 21: Fields of the table QRcodes*

The connection between the client and the server was made using the HTTP library called Volley, while the one between the server and the database uses JDBC (Java Database Connection).

## Evaluation

A pilot test was conducted in order to test the usefulness and quality of the proposed solution. Ten students of the University Carlos III of Madrid who had previously installed the application on their smartphones, took part in this test conducted in the university's library on June 12<sup>th</sup>, 2018. After a short introduction of the functionalities of the application they were challenged to complete the game.

Once they had ended the game, they were asked to fulfil a satisfaction survey where they could evaluate the design, the usefulness, the strengths and the weaknesses of the application. The results show a good acceptance of the system by the students. Besides, from their point of view, the game could be really useful for the new students and eight of them would download the app to use it daily.

The most repeated suggestion by the students was to introduce more levels, meanwhile the most appreciated points were the location system of the tables of the library and the originality of the levels of the game.

## Planification and budget

The development of this elaborated project was divided into smaller tasks to make it easier. The total duration of the project was approximately 10 months.

In respect of the budget of this system, we have to consider the software and hardware expenses as well as the cost of the manpower. All the software used for the

developing was open-source and free, so there is no cost of software to compute. The hardware used were two computers and one Android smartphone. Considering the average lifetimes of the computers and the smartphone as 6 years and 3 years respectively and the average cost of a computer and a smartphone as 900€ and 300€ respectively, we can conclude that the total investment in hardware is 333,33€.

Finally, for the cost of the manpower we have to consider every member of the team of development. As it shown in the Table 22, the cost of the manpower was 10900,50€

| Person                        | Category        | Cost per hour | Hours | Total    |
|-------------------------------|-----------------|---------------|-------|----------|
| Víctor Seoane<br>Mérida       | Newly qualified | 23,43€        | 420   | 9840,60€ |
| Iria Manuela<br>Estévez Ayrés | PhD             | 35,33€        | 30    | 1059,90€ |

*Tabla 22: Cost of the manpower*

So, the total budget for the design and implementation of the proposed system is 11233,83€.

## Conclusions and future work

There exist some activities to help the new students to know the library but sometimes these activities take place in the second or the third month of the academic year. This system does not intent to replace such activities but complement them.

As it can be seen in the evaluation's results, the system has had a good acceptance by the students who took part in the test and a large majority of them think it can be a useful tool for the new students, reaching the system's objectives.

This version of the system has some limitations which could be solved in the future. The most important of them is the need of using HTTPS to assure the security of the information. The decision of not implementing this protocol is really linked to the next limitation: the application has not been publicly distributed.

Other possible enhancements could be:

- Using GPS (Global Positioning System) for the location system instead of QR codes.
- Better graphic interface of the application.
- Ampliation of the game: new levels, enigmas and actions.
- Alternative routes of the game to allow multiple groups playing.
- Implementation of an iOS version to reach more users.



This system could even serve as a basis for the exploration of other areas and places of the university through the use of QR codes or for the implementation of a location server in the whole campus.



## Anexos

### Anexo 1: Pentagrama para el nivel 1.



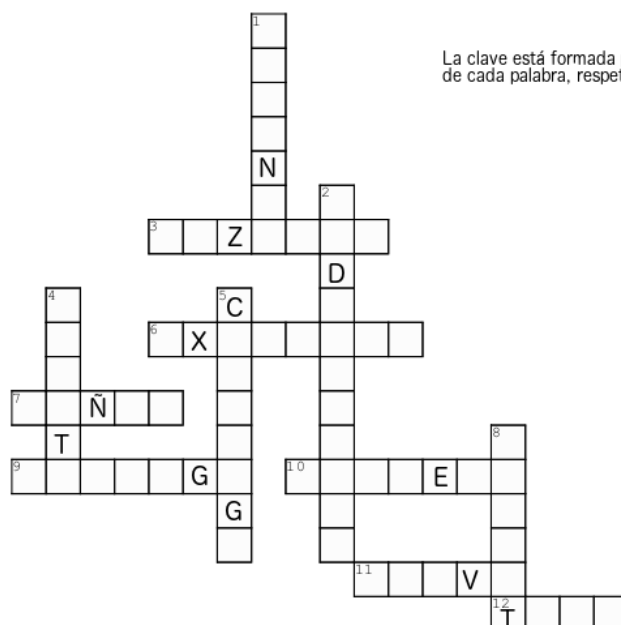
NOTAS: RE, RE, MI, RE, LA, SOL

### Anexo 2: Crucigrama para el nivel 5

#### Biblioteca UC3M

Complete el crucigrama para obtener la clave para el siguiente nivel

La clave está formada por la primera letra de cada palabra, respetando el orden 1-12



#### Horizontal

3. Disponible en todas las salas de estudio en grupo.
6. Época en la que solo los universitarios pueden acceder a la biblioteca de la UC3M.
7. Lugar que solo encuentras en la planta baja y tercera planta.
9. Uno de los días de la semana en los que la biblioteca no abre (salvo época de exámenes)
10. Planta donde está permitido hablar bajo.
11. Hora a la que suele abrir la biblioteca.
12. Número máximo de horas que puedes alquilar un portátil.

#### Vertical

1. Planta donde debes guardar silencio.
2. Dispositivos que puedes encontrar en el sótano, la planta baja y la tercera planta.
4. Aquí puedes encontrar aulas informáticas y audiovisuales.
5. Recurso que puedes consultar para encontrar un libro o recurso.
8. Lo necesitas para efectuar una reserva.

Soluciones:

- |                |             |
|----------------|-------------|
| 1- SEGUNDA     | 7- BAÑOS    |
| 2- ORDENADORES | 8- CARNET   |
| 3- PIZARRA     | 9- DOMINGO  |
| 4- SÓTANO      | 10- PRIMERA |
| 5- CATÁLOGO    | 11- NUEVE   |
| 6- EXÁMENES    | 12- TRES    |

CLAVE: SOPSCEBCDPNT

## Anexo 3: Encuesta de satisfacción de la aplicación

# uc3m | Universidad Carlos III de Madrid

### Encuesta sobre experiencia de uso de la aplicación para la biblioteca de la EPS

1. Valora la experiencia global de uso de la aplicación:

|                | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |                   |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-------------------|
| Muy deficiente | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy satisfactoria |

2. Valora la aplicación en cuanto a facilidad de uso:

|                    | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |               |
|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
| Muy poco intuitiva | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy intuitiva |

3. Valora la utilidad de las funcionalidades que ofrece la aplicación:

Juego de iniciación para nuevos alumnos:

|                  | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |          |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna utilidad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy útil |

Localización de mesas:

|                  | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |          |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna utilidad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy útil |

Acceso a servicios (horario, reserva de salas, préstamo interbibliotecario...):

|                  | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |          |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna utilidad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy útil |

Aplicación global:

|                  | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |          |
|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------|
| Ninguna utilidad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy útil |

4. Valora el diseño estético de la aplicación:

|                 | 1                     | 2                     | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     | 9                     | 10                    |               |
|-----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
| Muy poco visual | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Muy atractiva |

5. Describe lo que más te ha gustado de la aplicación

---

6. Describe algún aspecto a mejorar:

---

7. ¿Descargarías esta aplicación para usarla frecuentemente?

☐

Sí

☐

No

---

8. Algún otro comentario que consideres oportuno:

---